



Mathur, Suresh C. (1965) *Numerical studies in partial differential equations of elliptic type*. PhD thesis.

<http://theses.gla.ac.uk/3228/>

Copyright and moral rights for this thesis are retained by the author

A copy can be downloaded for personal non-commercial research or study, without prior permission or charge

This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the Author

The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the Author

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given

**NUMERICAL STUDIES IN PARTIAL DIFFERENTIAL
EQUATIONS OF ELLIPTIC TYPE being a Thesis presented
by SURESH CHANDRA MATHUR to the University of
Glasgow in application for the degree of DOCTOR OF
PHILOSOPHY.**

September 1965

Preface

The work presented in this thesis has been carried out under the supervision of Dr. D. C. Gilles.

The numerical results in Chapter V are to be published by Sneddon, Srivastav and Mathur.

I wish to offer my most sincere thanks to Dr. D. C. Gilles for his personal interest in me and for the encouragement and guidance I received from him.

I am also grateful to all members of the staff of the Computing Department for the willing cooperation they have shown me.

Suresh Chandra Mathur

CONTENTS

[Chapter 1 Treatment of curved boundaries in elliptic type partial differential equations, when
normal gradient is involved in boundary conditions.](#)

[Chapter 2 Mechnisation of the solution of Poissons's equation](#)

[Chapter 3 Numerical treatment of reentrant corner in Laplace equation in cylindrical polars
with axial symmetry.](#)

[Chapter 4 Extension of Milne's method to bi-harmonic equation.](#)

[Chapter 5 Numerical solution of two Fredholm type integral equations of second kind.](#)

[APPENDIX A](#)

APPENDIX B

[Index of reference](#)

INTRODUCTION

A variety of problems of physical interest are formulated in terms of elliptic partial differential equation. Well-known equations of this class are those of Laplace, Poisson and the bio-harmonic equation. The present work is the outcome of a study of some aspects of these equations. The problems included here are diverse in nature, and their study is numerical. Our aim is to explore numerical methods to handle each with adequate generality.

We devote a chapter to each problem. An outline of each chapter will now be presented.

In Chapter I we consider the case of curved boundaries in elliptic equations when the boundary conditions involve the normal gradient. Such boundaries have been considered by Fox (1950), Shaw (1950), Allen (1954), Viswanathan (1957), Varga (1957), Forsythe and Wasow (1959) and Greenspan (1965). Thus a variety of methods have been suggested for approximating the given partial differential equation at nodes, which lie adjacent to the curved boundary. In this chapter we present our approach towards this end. By including regional nodes as far as $h\sqrt{5}$ units from the typical node we developed relaxation patterns which have “almost” the same degree of accuracy as the formula for “irregular star”. Normally the resulting finite difference equations are diagonally dominant.

An ordering of nodes is suggested which reduces the band width of the coefficient matrix and also imparts Young’s property (A) to it.

Later, the technique is extended to treat boundaries of the “third type”, and an example considered by Allen, Fox, Southwell (1946) has been treated by it.

Chapter II presents the mechanization of the solution of Poisson’s equation. It is in partial answer to the big idea of developing a set of library programmes for the solution of similar equations, encountered so frequently in physical problems. A group of IBM 704 computer users, called SPADE, initiated this sort of project in 1957, and recently the KDF 9 Users group have worked up their own programme for the solution of Poisson’s equation. However these programmes cater only for a limited class of boundaries and layouts of the mesh over the region. On the other hand a general programme, which places no such restrictions, sounds a more welcome idea. Led by this thought we developed the programme we present here. The parametric difficulties with regard to the boundaries and the mesh have been overcome by combining appropriately with the user’s effort.

Chapter III presents the numerical treatment of the re-entrant corner in Laplace's equation in cylindrical polars with axial symmetry. Material of this nature seems to exist only with regard to the equations associated with the Laplacian operator. The works of Motz (1946), Jeffreys and Jeffreys (1950), Woods (1953) and Wilson (1962) are noteworthy in this connection. Apparently, however, we adopt the same principles as employed by the above authors. Accordingly, an appropriate series solution to our equation is worked out to take account of the function near the singularity. Further the direct methods of Motz and Jeffreys and Jeffreys are modified to work them iteratively. This procedure eliminates the prohibitive amount of algebra involved in these methods.

To illustrate our approach, we set up an actual problem, and work it out fully.

Subsequently, we deal with the error analysis.

In Chapter IV we consider the eigen value problem associated with the biharmonic equation. Milne (1957) has suggested a method for the numerical evaluation of eigen values associated with the ordinary differential equations with two-dimensional Laplacian operator.

In this chapter the same method is extended to the biharmonic equation in two dimensions. It is shown that using this method it is possible to improve the eigen values considerably.

Chapter V presents the numerical solutions of two Fredholm integral equations of the second kind. They are taken from the analytical works of Sneddon (1962a) and Srivastava (1963). The evaluation of the kernel is particularly interesting in both of them. Analytical treatment is offered to remove the singularity. In addition, the kernel of the second integral equation is of special computational interest. The integrand involves subtracting two functions of almost the same magnitude. The loss of significant digits is, therefore, serious. The technique adopted here to cope with this situation is of general applicability.

Subsequently, the two solutions are checked using appropriate methods. They are shown to be satisfactory.

This completes a brief description of the material presented here. Lastly, we may emphasize that wherever necessary we have included appropriate examples to illustrate our approach.

CHAPTER 1.

TREATMENT OF CURVED BOUNDARIES IN ELLIPTIC TYPE PARTIAL DIFFERENTIAL EQUATIONS, WHEN NORMAL GRADIENT IS INVOLVED IN THE BOUNDARY CONDITIONS.

1.1 Introduction

The partial differential equations of the elliptic type are, in general associated with curved boundaries, and one of the most frequently used methods of solving them is the one which solves a related problem in finite differences. The given bounded region is covered with a suitable lattice of points, called the nodal points or the nodes. The nodes within the given region may be classified as:

- (a) those lying on the given boundary or adjacent to it,
- (b) those lying within those of class (a).

At each node of class (b) the given partial differential equation is replaced by a suitable finite difference equation, while at nodes of class (a) the approximating finite difference equation incorporate the given boundary conditions in addition. The resulting system of finite difference equations is then solved simultaneously to give the wanted solution.

We thus notice that nodes of class (a) receive special treatment. In fact, no real difficulty is encountered in their treatment so long as the boundary cuts the lattice at nodal points only. But, equally so, the boundary may cut at non-nodal points as well. For example, a curved boundary may run across cutting the lattice at non-nodal points only. In such a situation, difficulty is experienced in developing reasonably accurate finite difference approximations to the given partial differential equation for (a) class nodes. It is the purpose of this chapter to present a general basis for developing suitable relaxation patterns for such difficult nodes, in problems in two dimensions.

With reference to figure 1, it is the point s like P and Q which have been referred to above as the difficult nodes. The point P

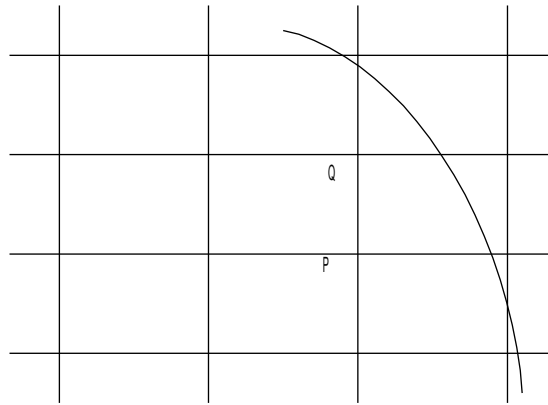


Figure 1.

has just one interrupted arm, while Q has two. When more than two arms are affected the net needs to be replaced by a finer one, and such a case has not been considered here at all.

Whatever be the given partial differential equation, the handling of the difficult nodes depends largely upon the nature of the given boundary conditions. Accordingly, it seems worthwhile classifying the boundary conditions first. A general classification of the boundary conditions would be:

- (1) When the wanted function is defined along the boundary;
- (2) When the normal derivative of the wanted function is defined along the boundary;
- (3) When the prescribed boundary condition is a linear combination of the above two conditions.

Condition (1) signifies simple Dirichlet boundary, and for such a boundary it is easy to derive relaxation patterns of reasonable accuracy. For example, for Nodes like P and Q, the formula for “Irregular star”, which is so commonly used in the solution of Poisson’s equation, is reasonably accurate – incorporating an error of the order of h for $\nabla^2 u$. We do not propose to consider this kind of boundary here. What is intended, in fact, is to develop formulae of “almost” the same degree of accuracy for boundaries of the kinds (2) and (3).

Such boundaries have been considered by Shaw (1950), Fox (1950), Allen (1954), Viswanathan (1957), Varga (1957), Forsythe and Wasow (1959) and Greenspan (1965). The general treatment adopted by Fox, Shaw, Allen, Forsythe and Wasow, in such a situation, favors the inclusion of “fictitious points”. However, the problem with regard to the elimination of such “fictitious points” is, sometimes, found to be quite enormous, and often, it is hard to ascertain the order of accuracy of the

resulting relaxation patterns. Viswanathan's method, which is also based on finite difference theory, has the same order of accuracy as the formula for "irregular star" for Poisson's equation. However, his relaxation pattern involved the radius of curvature of the boundary and also the first derivative of the given normal gradient with respect to the tangential direction at points where the boundary cuts the Lattice. The inclusion of these two parameters may, sometimes, discourage a programmer to use it freely in all sorts of boundaries. Varga's method (Which is also described in Fox (1962)), though not based on finite difference theory is comparatively easy to program, but an account of its order of accuracy cannot be obtained in all kind of boundaries. However, the basis presented here develops formulae free from "fictitious points", and further, the order of their accuracy is clearly indicated. Greenspan's approach is very much on the same lines as ours. A comparison would, therefore, be interesting. But we postpone it now, it is taken up exclusively in section 1.8; at the moment we proceed to present our approach.

1.2 The Present Approach

The basis of our approach is simple. What is suggested is that instead of including "fictitious points" in developing the relaxation patterns it is worthwhile including nodes from within the given region itself. It has been found that in most of the cases suitable regional nodes are never more than $h\sqrt{5}$ units away from the typical nodes. Even for a coarse mesh they can, therefore, be assumed to lie within the circle of convergence of the Taylor series of the wanted function about a typical node. Their inclusion does weaken the order of accuracy of the resulting relaxation pattern, but the price so paid is not so much as to disqualify them from consideration. For reasonably posed problems such nodes are easily available and the resulting relaxation pattern has "almost" the same order of accuracy as the formulae for "irregular star" for Poisson's equation.

To illustrate the approach, we will develop relaxation patterns for adjacent-to-the-boundary-nodes with boundary conditions of kinds (2) and (3).

1.3 Boundary Condition of Kind (2)

It is often referred to as the Neumann boundary and is commonly encountered in practice. We will illustrate our approach in relation to the solution of Poisson's equation in two independent variables x and y .

Accordingly, the problem is to determine a function w inside a region R when a function $g=g(x, y)$ is given within the said region such that

$$\frac{\partial^2 w}{\partial x^2} + \frac{\partial^2 w}{\partial y^2} + g = 0 \quad (1)$$

and that

$$\frac{\partial w}{\partial n} = k \quad (2)$$

Being given on the boundary of R .

Let (x, y) be any typical node, then using Bickley's notation (1948) let us denote it and its four adjacent nodes by 0, 1, 2, 3, 4 as in figure 2.

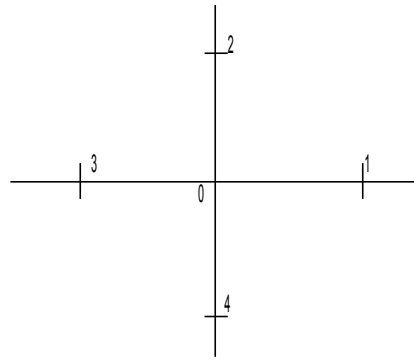


Figure 2.

On taking a square mesh of length h , we have $01 = 02 = 03 = 04 = h$.

If 0 is an interior node of R , i.e. if 0 is a node of class (b), then 0 and all its associated nodes 1, 2, 3, 4 lie within R and the finite difference representation of (1) at 0 is

$$w_1 + w_2 + w_3 + w_4 - 4w_0 + gh^2 = 0(h^4) \quad (3)$$

However this representation fails when the boundary cuts short any one or two of the four arms 01, 02, 03, 04. In such situations relaxation patterns are developed for adjacent-to-the-boundary-nodes by making use of both (1) and (2) and incorporating a suitable regional node as suggested above. It has been found that whether the number of interrupted arms is one or two, a single regional node is sufficient in both the situations.

1.4 One Interrupted Arm

Considering the node 0, (fig.3) let us assume that only one of the arms, say 01, has been interrupted at N by the Neumann boundary.

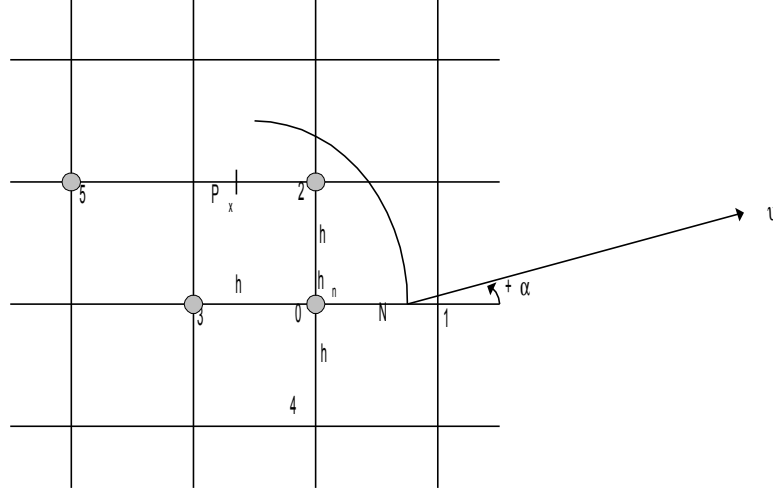


Figure 3.

Also, let $ON = h_n$ $0N = h_n$ where $0 \leq h_n \leq h$ and let the normal at N make an angle α with the interrupted arm in the positive sense. Then taking the node marked "5" as the regional node, suitable in this case, let us make our stand for a moment at P. Now assuming that the required function w has a Taylor Series at P and its circle of convergence encloses the points 2, 3, 4, 5 and N, we obtain

$$w_2 = w_p + \frac{h}{2} \left(\frac{\partial w_p}{\partial x} + \frac{\partial w_p}{\partial y} \right) + \frac{h^2}{8} \left(\frac{\partial^2 w_p}{\partial x^2} + \frac{2\partial^2 w_p}{\partial x \partial y} + \frac{\partial^2 w_p}{\partial y^2} \right) + O\left(\frac{h}{2}\right)^3 \quad (4)$$

$$w_3 = w_p - \frac{h}{2} \left(\frac{\partial w_p}{\partial x} - \frac{\partial w_p}{\partial y} \right) + \frac{h^2}{8} \left(\frac{\partial^2 w_p}{\partial x^2} - \frac{2\partial^2 w_p}{\partial x \partial y} + \frac{\partial^2 w_p}{\partial y^2} \right) + O\left(\frac{h}{2}\right)^3 \quad (5)$$

$$w_0 = w_p + \frac{h}{2} \left(\frac{\partial w_p}{\partial x} + \frac{\partial w_p}{\partial y} \right) + \frac{h^2}{8} \left(\frac{\partial^2 w_p}{\partial x^2} + \frac{2\partial^2 w_p}{\partial x \partial y} + \frac{\partial^2 w_p}{\partial y^2} \right) + O\left(\frac{h}{2}\right)^3 \quad (6)$$

$$w_4 = w_p + \frac{h}{2} \left(\frac{\partial w_p}{\partial x} - \frac{3\partial w_p}{\partial y} \right) + \frac{h^2}{8} \left(\frac{\partial^2 w_p}{\partial x^2} - \frac{6\partial^2 w_p}{\partial x \partial y} + \frac{9\partial^2 w_p}{\partial y^2} \right) + O\left(\frac{3h}{2}\right)^3 \quad (7)$$

$$w_5 = w_p + \frac{h}{2} \left(\frac{-3\partial w_p}{\partial x} + \frac{\partial w_p}{\partial y} \right) + \frac{h^2}{8} \left(\frac{9\partial^2 w_p}{\partial x^2} + \frac{6\partial^2 w_p}{\partial x \partial y} + \frac{\partial^2 w_p}{\partial y^2} \right) + O\left(\frac{3h}{2}\right)^3 \quad (8)$$

Also at the point P we have from (1)

$$\frac{\partial^2 w_p}{\partial x^2} + \frac{\partial^2 w_p}{\partial y^2} + g = 0 \quad (9)$$

Thus we have six equations in six unknowns.

$$w_p, \frac{\partial w_p}{\partial x}, \frac{\partial w_p}{\partial y}, \frac{\partial^2 w_p}{\partial x^2}, \frac{\partial^2 w_p}{\partial x \partial y} \text{ and } \frac{\partial^2 w_p}{\partial y^2} .$$

Hence on solving them we obtain

$$8w_p = 4w_2 + 2w_3 + w_4 + w_5 + 2gh^2$$

$$2h \frac{\partial w_p}{\partial x} = 2w_0 - 4w_p + 2w_2 + \frac{1}{2}gh^2$$

$$h^2 \frac{\partial^2 w_p}{\partial x^2} = -gh^2 - h^2 \frac{\partial^2 w_p}{\partial y^2}$$

$$h^2 \frac{\partial^2 w_p}{\partial x \partial y} = 4w_p - \frac{3}{2}gh^2 - w_5 - 2w^2 - w_4 \quad (10)$$

$$h^2 \frac{\partial^2 w_p}{\partial y^2} = 2w_p - 2w_0 - \frac{1}{2}w_3 + \frac{3}{4}w_4 - \frac{1}{4}w_5 - \frac{1}{2}gh^2$$

Further the point N we have

$$w_N = w_p + \left(\frac{h}{2} + h_n \right) \frac{\partial w_p}{\partial x} - \frac{h}{2} \frac{\partial w_p}{\partial y} + 0 \left(\frac{h}{2} + h_n \right)^2$$

Differentiating it w.r.t. x and y we obtain

$$\frac{\partial w_N}{\partial x} = \frac{\partial w_p}{\partial x} + \left(\frac{h}{2} + h_n \right) \frac{\partial^2 w_p}{\partial x^2} - \frac{h}{2} \frac{\partial^2 w_p}{\partial x \partial y} + 0 \left(\frac{h}{2} + h_n \right)^2$$

$$\frac{\partial w_N}{\partial y} = \frac{\partial w_p}{\partial y} + \left(\frac{h}{2} + h_n \right) \frac{\partial^2 w_p}{\partial x \partial y} - \frac{h}{2} \frac{\partial^2 w_p}{\partial y^2} + 0 \left(\frac{h}{2} + h_n \right)^2$$

Multiplying both by 2h we have

$$2h \frac{\partial w_N}{\partial x} = 2h \frac{\partial w_p}{\partial x} + 2h \left(\frac{h}{2} + h_n \right) \frac{\partial^2 w_p}{\partial x^2} + 0 \left[2h \left(\frac{h}{2} + h_n \right)^2 \right]$$

$$2h \frac{\partial w_N}{\partial x} = 2h \frac{\partial w_p}{\partial y} + 2h \left(\frac{h}{2} + h_n \right) \frac{\partial^2 w_p}{\partial x \partial y} - h^2 \frac{\partial^2 w_p}{\partial y^2} + 0 \left[2h \left(\frac{h}{2} + h_n \right)^2 \right] \quad (11)$$

Now considering the boundary conditions (2) we write

$$\begin{aligned} k_1 &= \frac{\partial w_N}{\partial v} \\ &= \frac{\partial w_N}{\partial x} \frac{\partial x}{\partial v} + \frac{\partial w_N}{\partial y} \frac{\partial y}{\partial v} \\ &= \frac{\partial w_N}{\partial x} \cos \alpha + \frac{\partial w_N}{\partial y} \sin \alpha \end{aligned}$$

or

$$2hk_1, Sec \propto 2h \frac{\partial w_N}{\partial x} + 2hm \frac{\partial w_N}{\partial y} \quad (12)$$

where $m = \tan \alpha$

Substituting the values of the derivatives from (11) we have form (12)

$$2hk_1 Sec \propto 2h \frac{\partial w_p}{\partial x} + 2mh \frac{\partial w_p}{\partial y} + 2h \left(\frac{h}{2} + h_n \right) \frac{\partial^2 w_p}{\partial x^2} - mh^2 \frac{\partial^2 w_p}{\partial y^2} + \left\{ 2mh \left(\frac{h}{2} + h_n \right) - h^2 \right\} \frac{\partial^2 w_p}{\partial x \partial y} + 0 \left[2h \left(\frac{h}{2} + h_n \right)^2 \right]$$

On substituting the values of the knowns from (10) and simplifying, we obtain,

$$4 \left(1 + \frac{h_n}{h} \right) w_0 - \left(1 + \frac{2h_n}{h} - m \right) w_2 - 2 \left(1 - \frac{mh_n}{h} \right) w_3 - \left(1 + m + \frac{2h_n}{h} + \frac{mh_n}{h} \right) w_4 - \frac{mh_n}{h} w_5 - \left(1 + \frac{mh_n}{h} + \frac{2h_n}{hgh^2} \right) - 2hk_1 Sec \propto 0 \left(\frac{3}{2} h \right) 3$$

denoting $\frac{h_n}{h}$ by ξ we can write it as

$$4(1 + \xi)w_0 - (1 - 2\xi - m)w_2 - 2(1 - m\xi)w_3 - (1 + m + 2\xi + m\xi)w_4 - m\xi w_5 - (1 + m\xi + 2\xi)gh^2 - 2hk_1 Sec \propto 0 \left(\frac{3}{2} h \right)^3 \quad (13)$$

It is the required relaxation pattern at 0.

We observe that the finite-difference equation (13)

- (a) is diagonally dominant,
- (b) is of non-negative type, (14)
- (c) imparts Young's property A to the matrix

provided

$$\begin{aligned} 1+2\xi &\geq m \\ m\xi &\leq 1 \end{aligned} \quad (15)$$

It is apparent that (15) are fairly reasonable conditions and in most of the problems they should be easily satisfied.

So far we considered the case when m was positive. Now let us consider the case when m is negative. Figure 4 illustrates the nature of the boundary curve for a negative m .

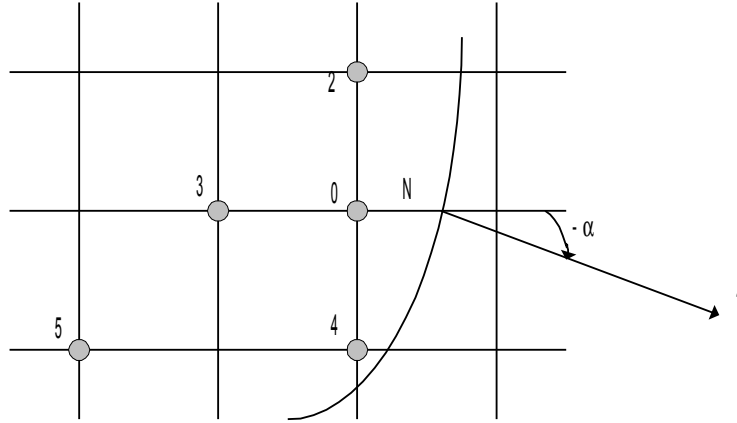


Figure 4.

In this case the suitable node is the marked "5" in the figure. The algebra involved here is almost the same as for positive m . Hence by a similar set of calculations we finally obtain difference equation at 0 to be

$$\begin{aligned} 4(1+\xi)w_0 - (1-m+2\xi)w_2 - 2(1+m\xi)w_3 - (1+2\xi+m)w_4 + m\xi w_5 - (1+2\xi-m\xi)gh^2 - \\ 2hk_1 \sec \alpha = 0 \left(\frac{3}{2}h \right)^3 \end{aligned} \quad (16)$$

We notice that equations (13 and (16) are the same, for if we set $-m = n$ in (16) we obtain

$$4(1+\xi)w_0 - (1+n+2\xi)w_2 - (2-2n\xi)w_3 - (1+2\xi-n)w_4 - nw_5 - (1+n\xi+2\xi)gh^2 - 2hk_1 \sec \alpha = 0 \left(\frac{3}{2}h \right)^3$$

which is the same as (13) except that the coefficients of w_2 and w_4 have been exchanged.

Hence, we are now in a position to generalize our results.

If we denote

$$\begin{aligned} P &\equiv 1 + 2\xi - |m| \\ Q &\equiv 2 - 2|m|\xi \\ R &\equiv 1 + |m| + 2\xi + |m|\xi \end{aligned}$$

Then according as the arms 01, 02, 03, or 04 is interrupted we have respectively at 0:

$$\begin{aligned} 4(1+\xi)w_0 - Pw_2 - Qw_3 - Rw_4 - |m|\xi w_5 - (1+2\xi+|m|\xi)gh^2 - 2hk_1Sec &\infty 0 \\ 4(1+\xi)w_0 - Pw_3 - Qw_4 - Rw_1 - |m|\xi w_5 - (1+2\xi+|m|\xi)gh^2 - 2hk_1Sec &\infty 0 \\ 4(1+\xi)w_0 - Pw_4 - Qw_1 - Rw_2 - |m|\xi w_5 - (1+2\xi+|m|\xi)gh^2 - 2hk_1Sec &\infty 0 \\ 4(1+\xi)w_0 - Pw_1 - Qw_2 - Rw_3 - |m|\xi w_5 - (1+2\xi+|m|\xi)gh^2 - 2hk_1Sec &\infty 0 \end{aligned} \quad (17)$$

While the conditions are

$$\begin{aligned} 1 + 2\xi &\geq |m| \\ |m|\xi &\leq 1 \end{aligned}$$

When m is -ve, then R and P exchange positions.

When $m = 0$ the equations reduce to a form which does not require the regional node "5". Also $P = R$ then.

The following scheme indicates the position of the regional node in the different cases.

		12		11	
	5			2	10
			3		1
			0		
	6		☆ 4		9

		7	8		

The Interrupted arm	Sign of m	Nodes used in relaxation position
01	+	0, 2, 3, 4, 5
01	-	0, 2, 3, 4, 6
02	+	0, 3, 4, 1, 7
02	-	0, 3, 4, 1, 8
03	+	0, 4, 1, 2, 9
03	-	0, 4, 1, 2, 10
04	+	0, 1, 2, 3, 11
04	-	0, 1, 2, 3, 12

1.5 Two Interrupted Arms

The formulation of the relaxation patterns is simpler when the star has two interrupted arms. As pointed out earlier only one regional node is needed in this case as well. But the number of suitable nodes available here is two. We are, therefore, in a comfortable position. In a reasonable problem one of them should certainly lie within the region. However, when both lay inside the region the one which does not increase the band-width of the resulting finite difference matrix may well be selected.

Let the boundary cut the Lattice as in figure 5. Let $ON = \xi h$, $OM = \eta h$

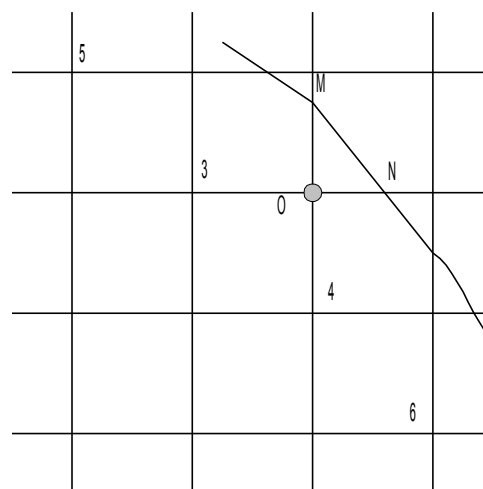


Figure 5

and the magnitude and direction of the normal gradient at N and M be k_1, α_1 and k_2, α_2 respectively. Further let $m_1 = \tan \alpha_1$, and $m_2 = \tan \alpha_2$, the two regional nodes available here are marked "5" and "6" in the figure. The signs of m_1 and m_2 are of little consequence here. Moreover the algebra involved in the derivation has the same set of steps as we encountered earlier. Hence it would be better if we write down the required difference equation at 0 straight away. So, accordingly as the point marked "5" is the regional node, the finite difference equation at 0 is

$$aw_0 - bw_3 - cw_4 - dw_5 - e = 0 \quad (18)$$

where

$$\begin{aligned} a &= 4(1 + \xi + n + m_1 n - m_2 \xi + m_1 m_2), \\ b &= 2[1 + (2 + m_2)n - (m_1 + 2m_2)\xi + m_1 m_2(1 - \xi - n) - 2\xi n(m_1 - m_2)], \\ c &= 2 + (4 + m_1)\xi + n(4m_1 - m_2) + 2\xi n(m_1 - m_2) + m_1 m_2(2 + \xi + n), \\ d &= m_1 \xi - m_2 n + 2\xi n(m_1 - m_2) + m_1 m_2(\xi + n), \end{aligned}$$

$$e = gh^2 \left[m_1 \left(1 + \frac{1}{2} \cdot \frac{1}{2} \cdot m_2 \cdot \frac{1}{2} \cdot m_1 \cdot \frac{1}{2} \cdot m_2 \right) + 2h \left(k_1 \cdot \frac{1}{2} \cdot m_2 \cdot \text{Sec } \alpha_1 + k_2 \cdot \frac{1}{2} \cdot m_1 \cdot \text{Sec } \alpha_2 \right) \right]$$

Next, if the point marked "6" is the regional node, the finite difference equation at 0 is still the same as equation (18), except that

w_5 is replaced by w_6

$$\begin{aligned} b &= 2 - (4m_2 - m_1)\xi + (4 - m_2)n + 2\xi n(m_1 - m_2) + m_1 m_2(2 + \xi + n), \\ c &= 2[1 + (2m_1 + m_2)n + (2 - m_1)\xi + m_1 m_2(1 - \xi - n) - 2\xi n(m_1 - m_2)] \end{aligned}$$

Other coefficients and the constant term remain the same.

It can be verified that in both the cases a, b, c, d should be positive for a reasonably posed problem. Hence equation (18) also enjoys the three properties enumerated in (14).

By an anticlockwise cyclic rotation of the suffixes of w in the second and third of (18) we can obtain finite difference approximations at 0 corresponding to stars with other pairs of interrupted arms.

16. An Example

To illustrate the merits of the above formulae, we consider an example. For this sort of boundary Vishwanathan's (1957) method is the most accurate of the existing methods. Hence to facilitate comparison with this method we solve the same problem which has been solved by him.

Hence we have to determine values of the harmonic function

$w = \log r = \frac{1}{2} \log(x^2 + y^2)$ inside the region ABCDE which is bounded by (figure 6).

$x = 0$ (straight line AE)

$y = 2$ (straight line ED)

$x = 3$ (straight line DC)

$y = 0$ (straight line CB)

$\frac{x^2}{4} + y^2 = 1$ (Elliptic arc BA)

with $\frac{\partial w}{\partial n}$ prescribed all along the border.

Here we have to solve Laplace's equation, hence $g = 0$ in (1) and in the formulae developed above. By taking $h = 0.25$ we have covered the region with the same lattice of points as Vishwanathan.

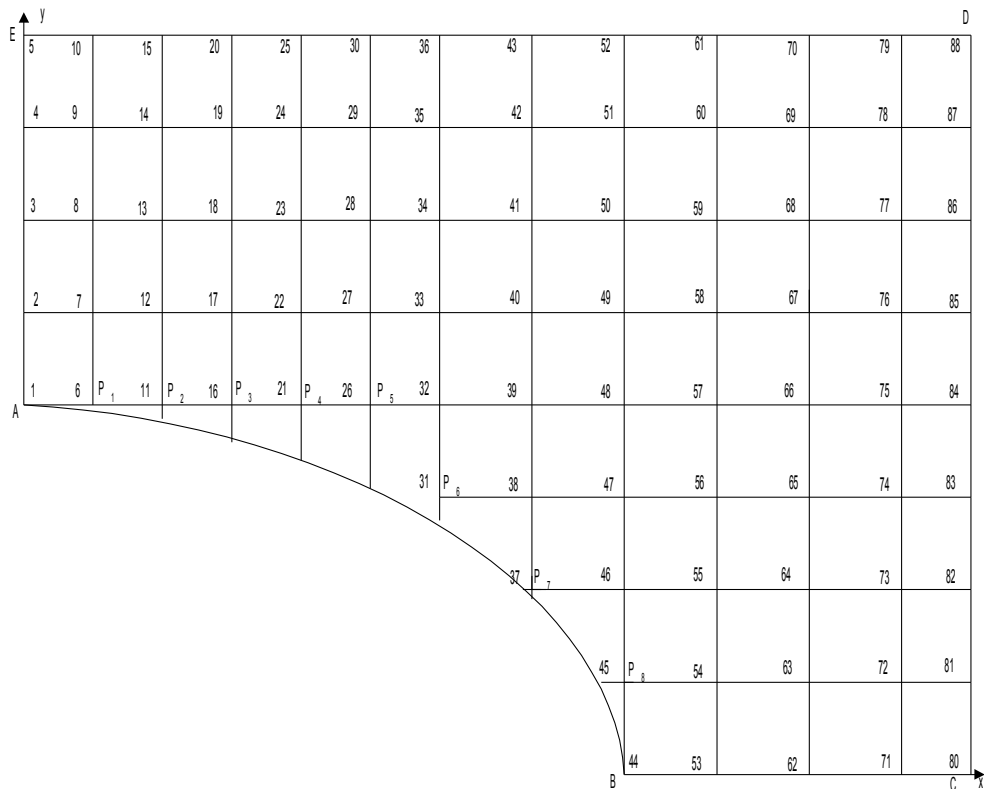


Figure 6.

All the eight nodes adjacent to the elliptic boundary AB have interrupted arms. Of these P₆ and p₇ have two interrupted arms while the remaining six nodes have only one each. In order to use the difference equations established above we must first calculate ξ , η , k_1 , k_2 , m_1 , m_2 for both P₆ and P₇ and η , k , m for the rest. All this data is presented in Table I.

On the basis of this table the difference equations can be written down at once. They are in Table II.

At the boundary nodes A, B, C, D, E the finite difference equations can be obtained from (18) by setting $\xi = \eta = m_1 = m_2 = 0$ and substituting appropriate values of k_1 and k_2 . The following table presents difference equations thus obtained at these nodes:

NODE S	Finite Difference Equations
A	$2w_1 - w_6 - w_2 = -.250$
B	$2w_{44} - w_{53} - w_{45} = -.125$
C	$2w_{80} - w_{81} - w_{71} = +.083$
D	$2w_{88} - w_{79} - w_{87} = +.09615383$
E	$2w_5 - w_{10} - w_4 = +.125$

At the remaining nodes on the boundary finite difference equations can be framed from (17) by setting $\xi = m = 0$, and assigning appropriate value to k_1 . For example, $k_1 = \frac{32}{65}$ at the node 10 (figure 6) the required finite difference equation from the second

TABLE I

NODE S	ξ	η	m1	m2	k1	k2
P1	0.03137303	-	- 0.06299408	-	- 0.96087057	-
P2	0.12701665	-	- 0.12909944	-	- 0.86256424	-
P3	0.29190076	-	- 0.20225996	-	- 0.74360247	-
P4	0.53589838	-	- 0.28867513	-	- 0.63394308	-
P5	0.87750100	-	- 0.40032038	-	- 0.54757784	-
P6	0.70849738	0.35424869	2.26778684	- 0.56694671	- 0.52756262	- 0.48937373
P7	0.07179677	0.06350833	1.15470054	0.90369611	- 0.46518704	- 0.46484081
P8	0.25403331	-	0.51639778	-	- 0.48139695	-

TABLE II

At node P_1 :
$1.03137303W_6 - 0.28192912W_{11} - 0.49901184W_7 - 0.24993800W_1 - 0.00049408W_3 = 0.12034690$.
At node P_2 :
$1.12701665W_{11} - 0.34988263W_{16} - 0.49180111W_{12} - 0.28123346W_6 - 0.00409944W_8 = 0.10871532$
At node P_3 :
$1.29190076W_{16} - 0.46127533W_{21} - 0.47048008W_{17} -$

0.34538539W ₁₁ -0.01475996W ₁₃ =0.09483251.
At node P ₄ :
1.53589838W ₂₁ -0.62879311W ₂₆ -0.42264973W ₂₂ - 0.44578041W ₁₆ -0.03867531W ₁₈ =0.08247861.
At node P ₅ :
1.87750100W ₂₆ -0.87665098W ₃₂ -0.32435923W ₂₇ - 0.58867040W ₂₁ -0.08782038W ₂₃ =0.07372807.
At node P ₆ :
0.24775907W ₃₁ -0.13488774W ₃₈ -0.05461757W ₃₂ - 0.05825376W ₂₇ =0.02463398.
At node P ₇ :
0.28752784W ₃₇ -0.13888845W ₄₆ -0.14305293W ₃₈ - 0.00558646W ₃₂ =0.02370771.
At node P ₈ :
1.25403331W ₄₅ -0.24791721W ₄₄ -0.43440888W ₅₄ - 0.53891166W ₄₆ -0.03279556W ₆₂ =0.06772430.

of (17) is

$$4w_{10} - w_5 - 2w_9 - w_{15} - \frac{16}{65}$$

It can be seen that the problem as formulated will not yield a unique solution. It will be general to the extent of an additive constant, which may be anything what-ever. Hence we have yet to impose an additional restriction to make it unique. We, therefore, use the fact that $\log r = 0$ to A. Hence using $w = 0$ to A as another boundary condition we now solve the problem as a mixed boundary value problem. Now we need not consider an equation at A, rather we need to set $w_1 = 0$ in the equations at the nodes P₁ and "2".

Having done this the results obtained are recorded in fig. 7. The true values and those of Vishwanathan are also given at each node for comparison.

1.6.1 Reducing the Band-width

It is true that due to the presence of the 'extra nodes' in our relaxation patterns, the band-width of the resulting matrix is, sometimes, much

larger than it would otherwise be. However, the effect of such nodes on the size of the band-width can be eliminated by scanning the region diagonal-wise. The one diagonal, which is, in a sense, parallel to the curved boundary, would serve as the best

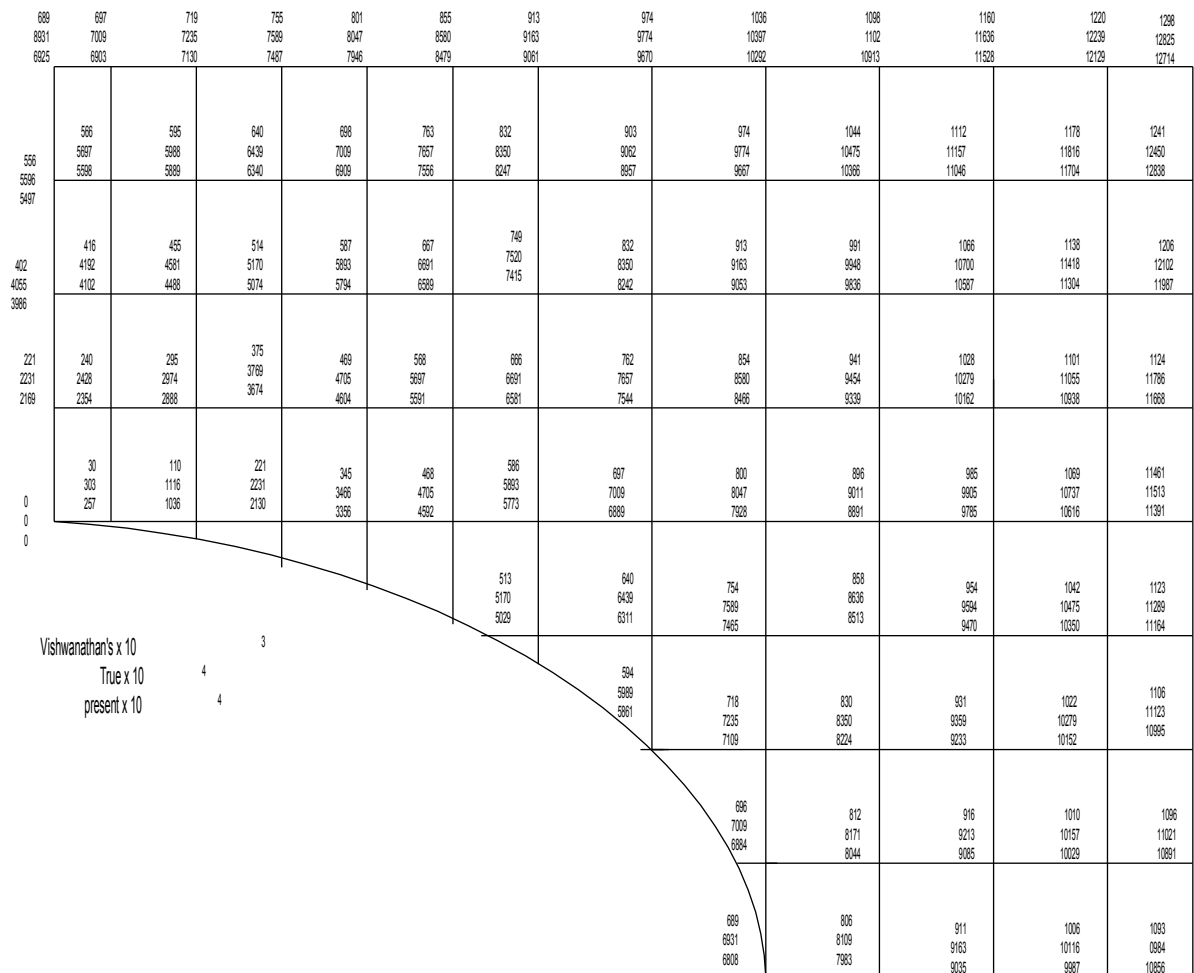


Figure 7.

pointer. The band-width would, then, be a function of only the maximum number of nodes along any diagonal. Unless the diagonal is really too large, the band-width would be considerably reduced by this technique. Moreover such diagonal-wise scanning of the region would also impart a tri-diagonal representation to the resulting matrix, which, as Property (A), is found useful when the linear system is solved by successive over-relaxation.

In the example considered above the facts are as follows:

- (1) If the nodes are taken in order, along the columns, that is, 1, 2, 3, 4, 5, 6, 7, 8,...; (fig 6), the semi-band-width of the matrix is 17.
- (2) If the nodes are taken in order, along the diagonals, that is 1; 6, 2; 11, 7, 3; 16, 12, 8, 4; 21, 17, 13, 9, 5; 26, 22, 18, 14, 10, 6; 31, 27, 23, 19, 15, 11, 7, 3, 1; (fig. 6), the semi-band-width is 10.

Thus a diagonal-wise ordering could reduce the semi-band-width from 17 to 10 in this case. It is a considerable achievement.

1.7. Boundaries of kind (3)

There is no difficulty in extending this technique to problems with boundaries of the kind (3). It may happen that when the boundary conditions are awkward, a suitable regional node may not always be available to lend the sort of sophistication as achieved in the finite difference equations in the last sections. In such a situation, however, we can make good by achieving greater accuracy instead. A nearby node, distant $h\sqrt{2}$ would be a better choice than the one distant $h\sqrt{5}$ from the typical node.

As an example of this kind, we consider the problem of determining the stresses in an axially symmetrical solid of revolution when it is subjected to a given axial symmetrical pressure. This problem has been considered by Allen, Fox, Southwell (1946). Employing cylindrical polar coordinates, with z-axis as the solid of revolution, and as the stresses involved are independent of θ , they have treated it as a two-dimensional problem in the z-r plane. The expressions for the four stress functions were derived by Southwell (1942) in terms of two functions ϕ and ψ , both dependent on r and z only, such that over the plane region of rotation, the simultaneous governing equations are

$$\frac{\partial^2 \phi}{\partial v^2} - \frac{1}{r} \frac{\partial \phi}{\partial r} + \frac{\partial^2 \phi}{\partial z^2} = 0 \quad (19)$$

$$\frac{\partial^2 \psi}{\partial r^2} - \frac{1}{r} \frac{\partial \psi}{\partial r} + \frac{\partial^2 \psi}{\partial z^2} = \frac{\partial^2 \phi}{\partial z^2} \quad (20)$$

While the conditions on the boundary are

$$\sin(z, v) \left[\frac{1}{r} \frac{\partial \phi}{\partial r} - \frac{1-\sigma}{r^2} \phi \right] = \nu r - \sin(z, v) \left(\frac{1}{r} \frac{\partial \psi}{\partial r} - \frac{\psi}{r^2 r} \right) - \frac{\cos(z, v)}{r} \frac{\partial \psi}{\partial z} \quad (21)$$

$$\sin(z, v) \frac{\partial \psi}{\partial z} = r \nu z + \cos(z, v) \frac{\partial \psi}{\partial r} \quad (22)$$

$$\psi = - \int r \nu z ds + \text{an arbitrary constant of integration} \quad (23)$$

Where v is the direction of the outward drawn normal at a point on the boundary, and νr and νz are given stresses at that point in the r and z direction respectively. σ being Poisson's ratio of the material of the solid of revolution.

Hence the problem is to determine ϕ and ψ subject to governing equations (19), (20) and boundary conditions (21,22,23) over the plane region whose axial rotation generates the given solid.

As usual we cover the region with a square mesh of length h and examine interrupted stars adjacent to the boundary. Allen, Fox, Southwell (1946) have found the case harder when the star has an interrupted arm parallel to z -axis. The technique of introducing “fictitious points” which they have employed here does not proceed well when they try to eliminate them at a later stage. However in treating it with the technique presented here, no great difficulty is encountered. As the boundary conditions are really awkward here, the resulting finite difference equation is not quite sophisticated, but it promises better accuracy anyway.

We propose to work out the steps fully.

With reference to figure 8, let the boundary cut the arm parallel to z -axis at Q , such that $OQ = \xi h$, we are now required to derive an appropriate finite difference equation at the node 0.

Eliminating $\frac{\partial \psi}{\partial r}$ between (21) and (22) we have at the boundary node Q .

$$\frac{1}{r} \frac{\partial \psi_Q}{\partial z} = vr \cos(z, v) + vz \sin(z, v) + \left[\frac{\psi_Q}{r_2} + \frac{1-\sigma}{r^2} \phi_Q - \frac{1}{r} \frac{\partial \phi_Q}{\partial r} \right] \sin(z, v) \cos(z, v) \quad (24)$$

Further vide boundary condition (23) we know ψ all along the boundary.

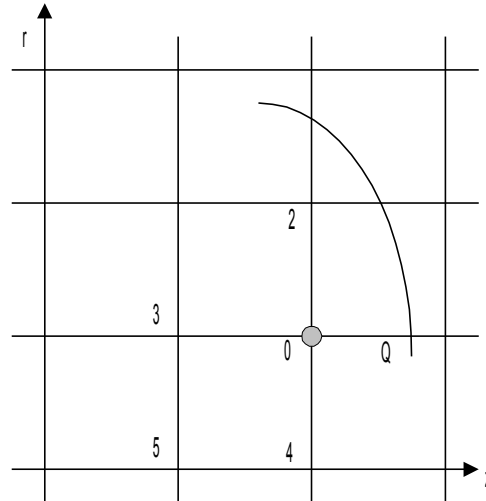


Fig. 8

Now considering the node marked “5” as the regional node agreeable here and assuming ϕ and ψ to possess Taylor’s series at 0 with nodes

2,3,4,5,Q within its circle of convergence, we obtain for ϕ , retaining terms upto the second derivative only.

$$\phi_Q = \phi_0 + \xi h \frac{\partial \phi_0}{\partial z} + \frac{1}{2} \xi^2 h^2 \frac{\partial^2 \phi_0}{\partial z^2} \quad (25)$$

$$\phi_2 = \phi_0 + h \frac{\partial \phi_0}{\partial r} + \frac{1}{2} h^2 \frac{\partial^2 \phi_0}{\partial r^2} \quad (26)$$

$$\phi_3 = \phi_0 - h \frac{\partial \phi_0}{\partial z} + \frac{1}{2} h^2 \frac{\partial^2 \phi_0}{\partial z^2} \quad (27)$$

$$\phi_4 = \phi_0 - h \frac{\partial \phi_0}{\partial r} + \frac{1}{2} h^2 \frac{\partial^2 \phi_0}{\partial r^2} \quad (28)$$

$$\phi_5 = \phi_0 - h \left(\frac{\partial \phi_0}{\partial z} + \frac{\partial \phi_0}{\partial r} \right) + \frac{1}{2} h^2 \left(\frac{\partial^2 \phi_0}{\partial z^2} + 2 \frac{\partial^2 \phi_0}{\partial z \partial r} + \frac{\partial^2 \phi_0}{\partial r^2} \right) \quad (29)$$

Also, rewriting (19) at 0

$$\frac{\partial^2 \phi_0}{\partial r^2} - \frac{1}{r} \frac{\partial \phi_0}{\partial r} + \frac{\partial \phi_0^2}{\partial z^2} = 0 \quad (30)$$

Solving (25), (26), (27), (28) we have

$$h^2 \frac{\partial^2 \phi_0}{\partial r^2} = \phi_2 - \phi_4 - 2\phi_0 \quad (31)$$

$$\frac{1}{2} h^2 \xi (1 + \xi) \frac{\partial^2 \phi_0}{\partial z^2} = \phi_Q + \xi \phi_3 - (1 + \xi) \phi_0 \quad (32)$$

$$h \xi (1 + \xi) \frac{\partial \phi_0}{\partial z} = \phi_Q - \xi^2 \phi_3 - (1 - \xi^2) \phi_4 \quad (33)$$

$$2h \frac{\partial \phi_0}{\partial r} = \phi_2 - \phi_4 \quad (34)$$

Substituting these values in (29) we get

$$h^2 \frac{\partial^2 \phi_0}{\partial z \partial r} = \phi_5 - \phi_0 + \frac{1}{\xi(1 + \xi)} [\phi_Q - \xi^2 \phi_3 - (1 - \xi^2) \phi_0] + h \frac{\partial \phi_0}{\partial v} - \frac{1}{\xi(1 + \xi)} [\phi_Q + \xi \phi_3 - (1 + \xi) \phi_0] - \frac{1}{2} (\phi_2 + \phi_4 - 2\phi_0) \quad (35)$$

again on substituting them in (31) we have

$$\phi_Q - (1 + \xi)^2 \phi_0 - \frac{1}{2} \xi (1 + \xi) \left(1 - \frac{h}{2r} \right) \phi_2 - \frac{1}{2} \xi (1 + \xi) \left(1 + \frac{h}{2r} \right) \phi_4 - \xi \phi_3 \quad (36)$$

Now differentiating (25) w. r. t 'r' we have

$$\frac{\partial \phi_Q}{\partial r} = \frac{\partial \phi_0}{\partial r} + \xi h \frac{\partial^2 \phi_0}{\partial z \partial r}$$

Substituting for $\frac{\partial \phi_Q}{\partial r}$ and $\frac{\partial^2 \phi_0}{\partial z \partial r}$, it is

$$\begin{aligned} \frac{1}{r} \frac{\partial \phi_Q}{\partial r} &= \frac{\xi}{rh} (\phi_5 - \phi_0) + \frac{1}{rh(1+\xi)} [\phi_Q - \xi^2 \phi_3 - (1-\xi^2) \phi_0] + \left(\frac{1+\xi}{r} \right) \frac{\partial \phi_0}{\partial r} - \frac{1}{rh(1+\xi)} [\xi \phi_3 + \phi_Q - (1+\xi) \phi_Q] - \\ &\frac{\xi}{2rh} (\phi_2 + \phi_4 - 2\phi_0) \end{aligned} \quad (37)$$

Replacing ϕ by Ψ in (25) and differentiating w. r. t. z

$$\begin{aligned} \frac{\partial \phi_Q}{\partial z} &= \frac{\partial \psi_0}{\partial z} + \xi h \frac{\partial^2 \psi_0}{\partial z^2} \text{ or} \\ h \frac{\partial \psi_0}{\partial z} &= \frac{1}{\xi(1+\xi)} [\psi_Q - \xi^2 \psi_3 - (1-\xi^2) \psi_0] + \frac{2}{1+\xi} [\psi_Q + \xi \psi_3 - (1+\xi) \psi_0] \end{aligned} \quad (38)$$

Substituting from (35), (36), (37), (38) in (24) we get

$$\begin{aligned} h.vr \cos(z, v) + h.vz \sin(z, v) + \psi_Q \left[\frac{h}{r_2} \sin(z, v) - \frac{1+2\xi}{r\xi(1+\xi)} \right] + \psi_0 \left(\frac{1+\xi}{r\xi} \right) - \frac{\xi}{r(1+\xi)} \psi_3 - \frac{\xi}{r} \phi_0 \sin(z, v) \cos(z, v) \\ - \left[\frac{h}{r^2} (1-\sigma)(1+\xi)^2 - \frac{\xi}{r} \right] \xi_0 \sin(z, v) \cos(z, v) - \left[\frac{h(1-\sigma)}{2r^2} \xi(1+\xi) \left(1 - \frac{h}{2r} \right) + \frac{1}{2r} \right] \phi_2 \sin(z, v) \cos(z, v) - \\ \left[\frac{h(1-\sigma)}{2r^2} \xi(1+\xi) \left(1 - \frac{h}{2r} \right) + \frac{1}{2r} \right] \phi^2 \sin(z, v) \cos(z, v) - \left[\frac{h(1-\sigma)}{r^2} \xi - \frac{\xi}{r} \right] \phi_3 \sin(z, v) \cos(z, v) - \\ \left[\frac{h(1-\sigma)}{2r^2} \xi(1+\xi) \left(1 - \frac{h}{2r} \right) - \frac{1}{2r} - \frac{\xi}{r} \right] \phi_4 \sin(z, v) \cos(z, v) = 0 \end{aligned} \quad (39)$$

Now adding (19) and (20) and replacing the derivatives by difference coefficients we get for the node 0,

$$\begin{aligned} \frac{\psi_Q}{r\xi(1+\xi)} - \frac{1+\xi\psi_0}{r\xi} + \frac{1}{2r} \left[\psi_4 \left(1 - \frac{h}{2r} \right) + \psi_4 \left(1 + \frac{h}{2r} \right) + \frac{2\psi_3}{1+\xi} \right] + \frac{1}{2r} \left[\phi_2 \left(1 - \frac{h}{2r} \right) + \phi_4 \left(1 + \frac{h}{2r} \right) - 2\phi_0 \right] = 0 \end{aligned}$$

$$\begin{pmatrix} 4 \\ 0 \\ 0 \end{pmatrix}$$

Adding (39) and (40) we finally get

$$\begin{aligned}
& h.v.r. \cos(z, v) + h.v.z. \sin(z, v) + \psi_Q \left[\frac{h}{r_2} \sin(z, v) \cos(z, v) - \frac{2}{r(1+\xi)} \right] + \frac{\psi_2}{2r} \left(1 - \frac{h}{2r} \right) + \frac{\psi_4}{2r} \left(1 + \frac{h}{2r} \right) + \frac{1-\xi}{r(1+\xi)} \\
& \phi_0 \left[\left\{ \frac{h(1-\sigma)(1+\xi)^2}{r^2} - \frac{\xi}{r} \right\} \sin(z, v) \cos(z, v) - \frac{1}{r} \right] - \\
& \phi_2 \left[\left\{ \frac{h(1-\sigma)}{2r^2} \xi(1+\xi) \left(1 - \frac{h}{2r} \right) + \frac{1}{2r} \right\} \sin(z, v) \cos(z, v) - \frac{1}{2r} \left(1 - \frac{h}{2r} \right) \right] - \\
& \phi_3 \left[\frac{h(1-\sigma)}{r^2} \xi - \frac{\xi}{r} \right] \sin(z, v) \cos(z, v) - \\
& \phi_4 \left[\left\{ \frac{h(1-\sigma)}{2r^2} \xi(1+\xi) \left(1 + \frac{h}{2r} \right) - \frac{1}{2r} - \frac{\xi}{r} \right\} \sin(z, v) \cos(z, v) - \frac{1}{2r} \left(1 + \frac{h}{2r} \right) \right] - \\
& \phi_5 \frac{\xi}{r} \sin(z, v) \cos(z, v) = 0
\end{aligned}$$

It is the required finite difference equation at the node 0.

This equation reduces to the one employed by Allen, Fox, Southwell (1946) for the rectangular region of rotation which they have considered in their numerical example.

1.8 **Concluding Remarks**

We will conclude this chapter by considering Greenspan's work on this subject, which is by far the latest in the field, and as his line of approach is very much similar to ours, things are all the more interesting.

In developing relaxation patterns for nodes adjacent to the Neumann boundary he also introduces extra nodes from within the given region.

We will consider his method M_2 , which is more accurate. In Fig.9, if 0 be the node in question whose arms 01 and 02 have been interrupted by the Neumann boundary AB,

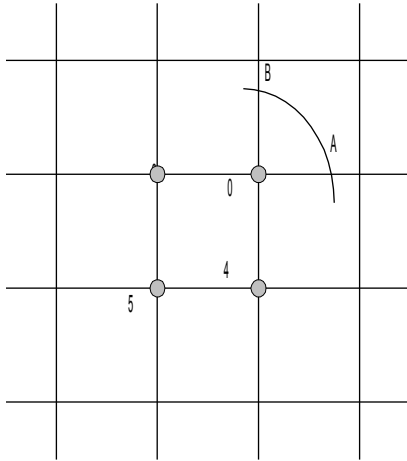


Figure 9

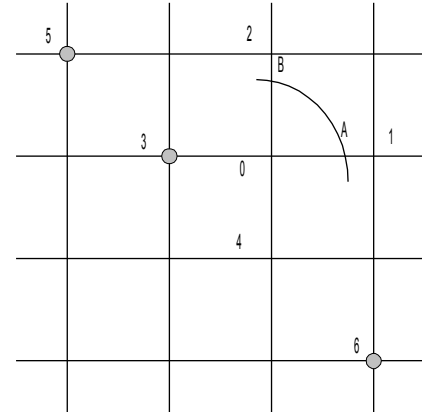


Figure 10

Then the extra node included by him is the one marked '5'. Instead in our case, we include either '5' or '6' (Fig. 10)

Naturally his relaxation pattern should be more accurate than ours, but at the same time there are certain other points which also deserve notice. As greenspan points out his relaxation pattern lacks properties (14) and further, the algebraic system generated by it may have either no solution at all, or may have more than one solution. None of these things is present in our relaxation pattern.

There is one more significant point in this connection. He dose not actually develop the relaxation pattern. For example, for the node 0, (fig. 9), he would like to incorporate as many as 3 equations in the linear algebraic system.

- (a) For 0 using the formula for "irregular star", imagining for a moment, that AB is a Dirichlet boundary.
- (b) For A, expressing the normal derivative at A in terms of the function values at B, 0, 4 and 5
- (c) For B, this time expressing the normal derivative at B in terms of function values at A, 0, 3, 4 and 5.

Effectively this means that the number of linear algebraic equations in the resulting system would, in his case, be always greater than the number of nodes inside the given region. However this could have been easily avoided by straightway developing the relaxation pattern.

From the above three equations after eliminating the function values at A and B.

CHAPTER II

MECHANISATION OF THE SOLUTION OF POISSON'S EQUATION

2.1 Introduction

Poisson's equation is so frequently encountered in physical problems that the need to having a set of Horary programmes to mechanise the whole process of its solution is gaining wider recognition. As early as 1957, a group of IBM 704 computer users, called SPADE, initiated this sort of project with the idea of mechanizing the solution of both the elliptic and parabolic partial differential equations. A comprehensive account of how " a very ambitious group of machine programmes was planned, and has been partially prepared" by them is given by Forsythe and Wasow (1960). The possibility of constructing a library of this sort of programme seems interesting to Fox (1962). The KDF 9 users group has gone far ahead. They have actually worked out a programme for the solution of Poisson's equation (1964).

In broad outlines, the programme of both the SPADE and the KDF 9. Users group seems to have much in common. It is in the sense that

(a) both generate the mesh automatically

(b) both use iterative methods for the solution of the system of algebraic equations.

They differ principally in their programming languages. While the SPADE programmes have been written in FORTRAN, the programme of the KDF 9 users group has been written in the KDF 9 user's code

The idea of allowing the machine to generate the mesh is indeed quite helpful to the user, for what is then expected from him, as input data, is just a few parameters, which indicate the way the mesh is desired to be laid over the region. The subsequent task, starting from the evaluation of details for each node, to the final output stage, is then the care of the programme itself. This sort of mechanization which takes in least input data, and consequently expects least effort from the user, appears brilliant indeed. However, it must be remembered at the same time that this achievement has been made only at the expense of restricting the nature of the boundary associated with the given region. The SPADE programme allow the boundary to be arcs of conic sections or segments of straight

lines only; while the programme of the KDF 9 users group does not allow the boundary to be other than straight lines or circular arcs. In addition to these major restrictions there are a few minor ones as well. For instance one such restriction which the programme of the KDF 9 users group imposes is that no horizontal mesh line may cut the boundary in more than four points.

It is easy to see that, in view of these restrictions the scope of these programmes appears rather limited. A general programme which could allow the boundary to be any curve whatever, should therefore be a more welcome idea. Led by this thought we developed one such programme, and we propose to present it in this chapter. To be precise it is written in the KDF 9 ALGOL language (publication 1002).

This programme places no restrictions on the nature of the boundary or on the lay out of the mesh over the region. However this generality has been achieved only by asking the user to supply his own information with regard to the boundary and the conditions thereon. In doing so the user is not expected to have any previous experience in programming. Simple ALGOL statements are all that is required. There are five procedures to cover this, and they are described in details in section 2.2.1

In order not to involve the user any further in programming, we have avoided the idea of generating the mesh through the machine. The user is therefore expected to cover the region with a mesh of his own choice and describe the type of each node in a coded form. The code consists of twenty integral numbers and they can cater for all the types of node permissible. Along with this code description of each node is associated a few more parameters. All these numbers together form the required data. It is all described in details in section 2.2.1

The five procedures and the data is, therefore, all that our programme required from the user. Thus by making a reasonable compromise with the user we have overcome the parametric difficulties which hold back the mechanization process from acquiring a generality.

There is yet another point where our programme differs from the programmes mentioned above. It is with regard to the method of solution of the system of finite difference equations. The above programmes use iterative methods, but we have adopted a direct method instead. Iterative methods are indeed advantageous for solving very large sparse systems, but for problems of moderate size, direct methods are preferable. They yield reasonably accurate solutions in a finite number of steps, while iterative methods may often require a very large number of iterations to yield the same degree of accuracy.

The fact that our programme is written in ALGOL imparts it yet another form of generality. As the ALGOL language is being internationally standardized, it is always possible to modify the programme to use it on any machine having an ALGOL translator.

2.2.1 Outlines of the programme

Our programme assumes that the given region over which Poisson's equation is being considered is regular and none of its boundary points has any type of singularity either.

Further the programme assumes that the region is covered with a square mesh.

The boundary may be segment of any curve, and the condition thereon may be either Dirichlet (Function value specified) or Neumann (Normal derivative specified).

In case of Dirichlet boundary the finite difference equivalence to poisson's equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = g(x, y)$$

is provided by the formula

$$\frac{2u_1}{h_1(h_1 + h_3)} + \frac{2u_2}{h_2(h_2 + h_4)} + \frac{2u_3}{h_3(h_3 + h_1)} + \frac{2u_4}{h_4(h_4 + h_2)} - 2u_0 \left(\frac{1}{h_1 h_3} + \frac{1}{h_2 h_4} \right) = g(x, y)$$

Where u_0, u_1, u_2, u_3, u_4 are the function values in the accepted sense and h_1, h_2, h_3, h_4 are interrupted arms. Of course h is the mesh length. In case of a regular interior node we have

$$h_1 = h_2 = h_3 = h_4 = h,$$

and the above formula reduces to well-known form, viz.

$$u_1 + u_2 + u_3 + u_4 - 4u_0 = h^2(g, y).$$

The programme assumes that not more than two arms of any node in the region have been interrupted by the boundary. If however this is not the case, the region need be covered with a finer mesh. A check on this is necessary because otherwise the programme will step out giving a failure signal.

In case of Neumann boundary, the finite difference equations are produced according to the patterns described in Chapter I. We have no intention of reproducing them here. We will however point out that before using the programme the user may make sure that the “extra” nodes demanded by those relaxation patterns are actually available inside the region, otherwise the programme will step out of the failure exit. Again the programme assumes that a maximum of only two arms of any node in the region have been interrupted by the Neumann boundary, and further, these arms are never apposite arms of a star. For instance a node with its first and third arms interrupted is not acceptable. Likewise any node with its second and fourth arms interrupted is not accepted either.

2.2 The programme

The programme comprises of three parts

1. The band formation
2. The band trimming
3. The solving process
4. The output part

Henceforth, we will call them as parts I, II, III and IV respectively. We now proceed to describe the function of each part.

2.2.1 Part I (The band formation)

This is by far the most important part of the whole programme . By itself it does a variety of jobs, and at the same time, prepares necessary data for the following three parts . As the user is directly concerned with this part, and with no other part at all, we propose to describe it in some detail.

To begin with, the user is required to plant the axes of reference in the plane of the given region. This he may do arbitrarily, keeping in view his convenience for expressing the nodes and the boundaries in terms of simple analytical geometry. Next with reference to these axes, he is required to write five ALGOL procedures, which hold information regarding

the Poisson-term, the nature of the boundaries and the conditions thereon. The five procedures are :

- i) POISSON
- ii) DESCRT
- iii) DIRI
- iv) NEUM
- v) DERV

In effect the programme assumes that the user has the poisson's equation written in the form

$$\frac{\partial^2 \phi}{\partial x^2} + \frac{\partial^2 \phi}{\partial y^2} = g(x, y) \quad (1)$$

and the way the programme sets up the finite difference equations is in the matrix notation

$$Ax = b \quad (2)$$

Where , A is the coefficient matrix, b the right hand side and x the wanted vector.

Accordingly, it is in the context of (1) that the user writes the procedure POISSON to define the Poisson-term g in his case. There are three formula parameters in this procedure, let us call them f_1, f_2 and f_3 . The first two parameters , viz. f_1 and f_2 refer to the x and the y coordinates of the node in equation. The third viz. f_3 stands for the value of the Poisson-term g at that node. Hence , in this procedure user makes a single ALGOL statement, saying

$$f_3 := g(f_1, f_2)$$

The procedure DESCRT defines the boundaries in terms of two-dimensional analytical geometry. There are five formal parameters in it. Let us call them f_1, f_2, f_3, f_4 and f_5 .

The first three viz. f_1, f_2 and f_3 are called by value, while f_4 and f_5 are called by name . f_1 and f_2 stand for the x and y coordinates of the adjacent-to-the-boundary-nodes and f_3 denotes the serial number of the boundary involved (the boundaries are arbitrarily numbered by the user). f_4 and f_5 stand for the x and y coordinates of the points in which the boundary cuts the lattice.

end;

where b_1 is the serial number of the boundary involved here.

The next procedure is DIRI. It is meant to provide the function value on all the Dirichlet boundaries of the region. As told already the boundaries are arbitrarily numbered by the user and on each one of them the function values are described in simple ALGOL statements. There are four formal parameters in it. Let us call them f_1, f_2, f_3 and f_4 . The first three parameters, viz. f_1, f_2 and f_3 are called by value, while f_4 is called by the name. f_1 and f_2 refer to the x and y coordinates of the points where the boundary cuts the lattice, and f_3 is the serial number assigned to this boundary. f_4 is of the type real and stands for the function value. Hence in this procedure the user makes statements, saying

$$\begin{aligned} \text{if } f_3 = b_1 \text{ then } f_4 &:= F_1; \\ \text{if } f_3 = b_2 \text{ then } f_4 &:= F_2; \end{aligned}$$

And so on. Here b_n is the serial number of the boundary and F_n is the function value on it.

The procedure NEUM is the analogue of the procedure DIRI. It defines the value of the normal derivative at each of the Neumann boundaries of the given region. Again there are 4 formal parameters in it ; let us call them f_1, f_2, f_3 and f_4 . the first three parameters, viz. f_1, f_2 and f_3 have exactly the same references as in the procedure DIRI, and therefore , they hardly need to be explained any more. However , f_4 stands for the normal gradient this time. But the ALGOL statements in this procedure , are given similar to the ones in the procedure DIRI. For example , we say

$$\begin{aligned} \text{if } f_3 = b_1 \text{ then } f_4 &:= N_1; \\ \text{if } f_3 = b_2 \text{ then } f_4 &:= N_2; \end{aligned}$$

etc. ,

Where b_n denotes the serial number of the boundary and N_n denotes the normal gradient on it.

The least procedure is DERV. It provides the angles which the tangents to the boundaries make with the axis of x, measured in the positive sense. Hence , it describes the value of the angle.

$$\arctan(dy / dx)$$

At each boundary on which the normal gradient is prescribed. Again, there are four formal parameters in it, and the first three have exactly the same meaning as in DIRI or in NEUM. f_4 stands for the angle this time. The ALGOL statements are now

$$\begin{aligned} &\text{if } f_3 = b_1 \text{ then } f_4 := A_1 ; \\ &\text{if } f_3 = b_2 \text{ then } f_4 := A_2 ; \end{aligned}$$

Where b_n denotes the serial number of the boundary involved here, and A_n denotes the angle which the tangent to the boundary makes with x-axis in the positive sense. It is clear that, depending upon the nature of the boundary, the expression for A_n will involve the parameters f_1 and f_2 .

In addition to the five procedures mentioned above, the user is also required to furnish a small piece of data. Essentially, it is a row-wise (or column-wise) description of the types of the nodes he has in the region.

An assortment of twenty types of nodes is permissible. It is divided into four categories :

- (i) 0 ;
- (ii) 1 ;
- (iii) -1, -2, -3, -4, -12, -13, -14, -23, -24, -34 ;
- (iv) -11, -22, -33, -44, -1122, -2233, -3344, -1144.

We now proceed to describe each category.

An O-type node is a node on a Dirichlet boundary. No difference equation corresponding to this node is set up.

A 1-type node is an interior node. It has no interrupted arm, all its four neighboring nodes exist the region.

The category (iii) refers to nodes with arms interrupted by one or two Dirichlet boundaries. For example, the node P (fig. 2) will be of the type -1, and Q (fig. 3) of the type -23. The idea is to call the missed

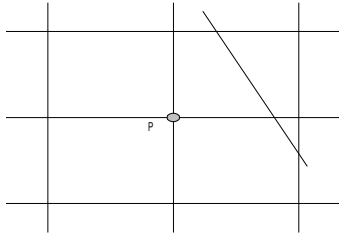


Figure 2

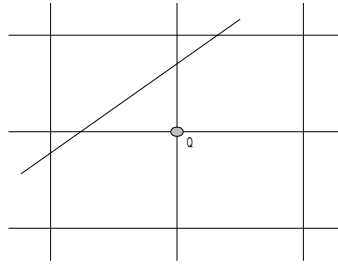


Figure 3

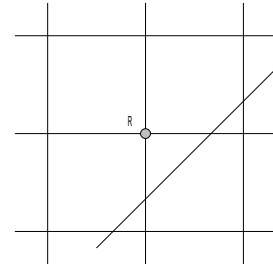


Figure 4

Out node or nodes, remembering that a node like R (fig. 4) would be called -14 and not -41.

As already pointed out, a description of the nodes is not the only thing which forms the data. It consists of certain other details as well. To facilitate full description, we consider the data to be made up of:

1. The Parameters
2. The Data Table.

The Parameters are always twelve in number, while the size of the Data Table depend upon the nature of the region. Although the Parameters are read in first and the Data Table follows, they cannot be determined until the Data Table is ready. We will call the twelve Parameters as P1, P2. ...P12 in order to proceed to illustrate the method for data preparation with reference to certain typical examples.

Example 1 Let u satisfy Laplace's equation

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = 0$$

in a quadrant of a unit circle

$$x^2 + y^2 = 1$$

with the boundary conditions that

(i) $u = 0$ on $y = 0$

(ii) $u = y$ on $x = 0$ and

(iii) $\frac{\partial u}{\partial y} = y \frac{\partial u}{\partial y}$ on the circular arc. To find u .

Let us cover the region with a square mesh length = 0.25, and number the boundaries

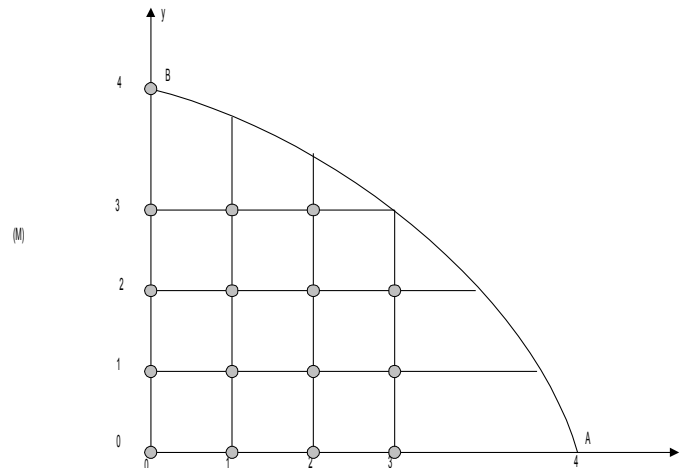


Figure 5

OA, AB, BO as 1, 2, 3 respectively. Accordingly, the first five procedures are :

Procedure POISSON (X,Y, Poisson term) : value X,Y;

real X,Y, Poisson term ;

Poisson term := 0 ;

Procedure DESCRT (X,Y, boundary, x,y) ;

Value X,Y, boundary ;

Real X,Y, x,y; integer boundary;

Begin if boundary = 2 then begin x := X; y := Y; and;

if boundary = 2 then begin x := sqrt (1,0-(Y x Y) ;
Y := sqrt (1,0-(X x X));

End ;

if boundary = 3 then begin x:= 0 ; y := Y and ;

End ;

Procedure DERV (X,Y, boundary, angle);

Value X,Y, boundary ;

Real X,Y, angle ; integer boundary ;


```

      Begin if boundary = 1 then angle : = 0;
            if boundary = 2 then angle : = 3.14159265-arctan
(X/Y);
            if boundary = 3 then angle : = 1.57079633;
      End;

```

Procedure DIRI (X,Y, boundary,d);

```

      Value X,Y,boundary;
      Real X,Y,d; integer boundary ;
      Begin if boundary = 1 then d : = 0 ;
            if boundary = 3 then d : = y;
      End ;

```

Procedure NEUM (X,Y,boundary,n);

```

      Value X,Y, boundary ;
      Real X,Y, n; integer boundary;

      Begin if boundary = 2 then n : = y;
      End ;

```

Now, having read in the procedures, we have yet to read in the data. As already pointed out , we will prepare the data table first and the parameters next, although they are fed into machine in the reverse order.

All the nodes marked with a thick dot are the Dirichlet boundary nodes and are , therefore , of the type 0. The remaining nodes , marked with crosses, are of the types specified in categories (ii), (iii) and (iv) above. All the nodes, irrespective of their category , are described in the Data Table. But the user has to decide the way he prefers their description. There are four ways of describing these nodes;

- (a) row-wise, upwards, starting from the lowest;
- (b) row-wise, downwards, starting from the uppermost ;
- (c) column-wise, from left to right;
- (d) column-wise , from right to left.

Let us choose the way specified in, say, (A). The form of the Data Table for this problem is;

1;	14;	0;1;	0;4;	0;0;0;0;0;	1;1;1;1;1;
1;	9;	1;1;	0;3;	0;2;-11;	3;2;
1;	10;	2;1;	0;3;	0;2;-1122;	3;2;2;(m)
1;	11;	3;1;	0;2;	0;-22;-1122;	3;2;2;2;
1;	6;	4;1;	0;0;	0;	3;

We will now explain this table. It consists of five arrays, and they are all constructed in the same fashion. A description of any one of them, will therefore, cover for the rest. Let us take up the middle array marked (M) and explain the meaning of each of its elements. This array refers to the row marked (M) in figure 5.

The first element is 1. This indicates that there is only one row of this nature. None of its adjacent rows is its copy.

The next element is 10. It is just the count of the elements which follow it in the array (M).

The third element is 2. It is the ordinate of this row. The unit of measurement being the mesh length. Obviously, if the row is towards y-increasing this integer is positive. If the row is on the -ve side of the -y axis, then this number is negative.

The fourth element is 1. It means that this row is just one continuous whole. If the given region is pierced with holes then the whole row would be broken up in to segments. The number occurring here is actually the count of these segments.

The fifth and the sixth elements refer to the abscissa of the first and the last nodes of this row in units of mesh lengths.

The following three elements indicate the types of the nodes in this row. The first node is of the type O. The next two nodes are of the type 1, and since, both the nodes are adjacently placed, we can economise by just writing 2, in place of 1; 1;. The fourth node has its first and the second arm interrupted by the Neumann boundary AB; accordingly, it is of the type - 1122.

The last three elements describe the boundaries affecting the nodes of this row. The first node of this row lies on OB which is our boundary no. 3. Hence, 3 appear as the first of these three elements. The remaining two elements, viz. 2; 2; refer to the boundary no. 2, i.e. AB which has affected the two arms of the last node of this row. It may be emphasized here that whenever two arms are interrupted, both the boundaries need be declared.

This completes the description of the array (M) pertaining to the row (M) in figure. 5. The same applies to the rest of the rows of the above Data Table, and any more explanation is, therefore, unnecessary. We can now proceed ahead to the parametric part of the data. The 12 parameters written with the help of the above Data Table are as follows:

$P_1 = 5$; (Total no. of rows in the given region).

$P_2 = 5$; (No. of nodes in a row which has a maximum of them).

$P_3 = 3$; (No. of nodes, not of type O, in a row which has a maximum of them).

$P_4 = 9$; (total no. of nodes in the region which are of type O).

$P_5 = 1$; (As an indication of whether we are reading the region in the positive or the negative direction we use +1 or -1. Whenever we are reading row-wise upwards, or column-wise from left to right, $P_5 = +1$. Alternatively, a row-wise downwards read, or a column-wise right to left read assigns $P_5 = -1$).

$P_6 = 1$; (When we read row-wise $P_6 = 1$, and when we read column-wise $P_6 = 0$. Here we are reading row-wise, hence $P_6 = 1$).

$P_7 = 0$; (A row-wise read means that $P_7 = 0$, while a column-wise read means that $P_7 = 1$. Here, we are reading in row-wise, hence $P_7 = 0$).

$P_8 = 0.25$; (It is the mesh length).

$P_9 = 14$; (It is the maximum number appearing in col.2 in Data Table).

$P_{10} = 5$; (Total no. of arrays in the Data Table).

$P_{11} = 5$; (The function values are output in exact geometric positions of the corresponding nodes. A maximum of eight columns is permissible. If the

number of columns, in the given region, exceeds eight, an automatic page change takes place. Hence the number mentioned in this position should never be greater than 8. in our problem we have only five columns, hence it is 5 here).

$P_{12} = 0$; (Both the fixed and the floating form of outputs are possible. A “0” indicates a fixed format $-n \text{ ddddd}$, while a “1” indicates floating format $-d.\text{dddd}_{10} \neq nd$).

Thus we have furnished a full description of all the 12 parameters. With the procedures already given, the information, necessary for the solution of this problem is now complete. No further detail is required.

Before moving to the next example, we would like to point out certain subtle points with reference to the above example. When the nodes are read in row-wise, the width of the band is a function of the maximum number of non-0-type nodes in any one row. On the other hand, if we read in column-wise the band width is a function of the maximum number of non-0-type nodes in any one column. In the particular example above the maximum number of nodes, row-wise and column-wise, is the same viz. 3. But in most of the problem they are different, and naturally an advantage is gained by choosing the one which is least. The shorter the band the better it is. It is why, the user may sometimes, prefer to read in column-wise. As an illustration of how the data, for the above example could be read in column-wise, from right to left, we have the following form.

The Parameters

5; 5; 3; 9; -1; 0; 0.25; 14; 5; 5; 0

The Data Table

1; 6; 4; 1; 0;0; 0; 1;

1; 11; 3; 1; 0; 2; 0; -11; -1122; 1; 2; 2; 2;

1; 10; 2; 1; 0; 3; 0; 2; -1122 1; 2; 2; 2;

1; 9; 1; 1; 0; 3; 0; 2; -22; 1; 2;

1; 14; 0; 1; 0; 4; 0; 0; 0; 0; 0; 3; 3; 3; 3;

In the context of the explanation already given, it is easy to understand how the Parameters and the Data Table have been written out in the above form. The procedures hold good whatever way the data is read in.

Example 2 Let the stress-function u satisfy Poisson's equation.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} = -2$$

Inside an H-section girder, (fig.6) $u=0$ on the boundary. Solve for u inside the region. But, in order to bring into focus certain aspects of the Programme, which could not be illustrated by Example 1, we propose to consider the whole of the H-shaped region ABCDEFGHJKLMN.

Having covered it up with a suitable net of mesh length h , let use

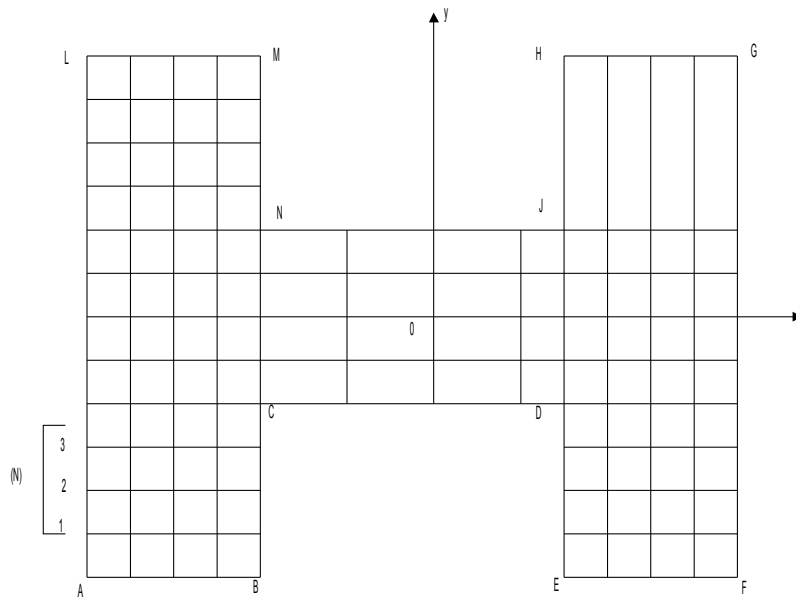


Figure 6.

Plant the axes of reference as shown in figure 6, and number the boundaries in the following order.

Name of the boundary	Number allocate to it
AB and EF	1
BC	2
CD	3
DE	4

FG	5
GH AND LM	6
HJ	7
JN	8
NM	9
LA	10

We now decide that we will read in row-wise, from ABEP to LMHG, i.e. towards y - increasing. Accordingly the Parameters and the Data Table are:

13; 13; 11; 62; 1; 1; 0; h; 26; 7; 8; 0;
1; 26; -6;2; -6;-2;2;6; 0;0;0;0;0;0;0;0;0;0;0; 1;1;1;1;1;1;1;1;1;1;
3; 16; -5;2; -6;-2;2;6; 0;3;0;0;3;0; 10;2;4;5;
(N)
1; 20; -2;1; -6;6; 0;3;0;0;0;0;0;3;0; 10;3;3;3;3;3;5;
3; 9; -1;1; -6;6; 0;11;0; 10;5;
1; 20; 2;1; -6;6; 0;3;0;0;0;0;0;3;0; 10;8;8;8;8;8;5;
3; 16; 3;2; -6;-2;2;6; 0;3;0;0;3;0; 10;9;7;5;
1; 26; 6;2; -6;-2;2;6; 0;0;0;0;0;0;0;0;0;0;0; 10;10;10;10;10;10;10;10;10;10;

Here we need not explain the parameters. They are written in the manner already explained in example 1. We will only discuss the second array marked (N) in above table. This array corresponds to the three rows, collectively marked (N) in fig. 6.

The first element of this array is 3. It suggests to the machine that there are three adjacent rows (marked 1,2,3 in fig. 6) which have the same description. In such a situation, a description of the first one of them is given. Hence we will describe the one marked "1" here.

The second element is 16. It is the count of the elements which follow it in the array (N).

The third element , which is -5 is the ordinate of the first row, in units of mesh length.

The fourth element is 2. It indicates that each of these rows has two segments.

The fifth and sixth elements are -6 and -2. They are the abscissa of the first and the last nodes of the first segment. The seventh and eighth elements, which are 2 and 6, are, similarly, the abscissa of the first and last nodes of the second segment. The unit of measurement being the mesh length.

The rest of the elements have nothing new to be explained about. They bear the same explanation as in Example 1.

We thus notice that instead of writing 3 arrays for the 3 rows, we have economized by just writing the details of only the first of them. Another things worth noticing is the way the different segments of a row have been described.

2.2.2 Part II (the band trimming)

The band formed by part I is written down on a magnetic tape. It may well be, that the band so formed may have quite a few columns of zero elements on both sides of it. Such zero-columns should simultaneously be dropped from both the sides, because, while it does not affect the analysis, it does help a great deal by way of reducing the band width.

Accordingly, the part of the programme inspected the band as formed by part I, and, if necessary, trims it from both the sides. The finished band, as produced by this part, has the minimum band width possible.

The crude version of the band is, then, over-written by finished one.

The user has nothing to do with this part directly, because, as soon as part I is over, this part steps in automatically.

2.2.3 Part III (the solving process)

The finished version of the band equation, as produced by part II, is then solved against the right hand side produced by part I. This solution over-writes the band this time.

The simple elimination method without interchanges is the one used here in the solution of the band equations. The given band A is factorised into lower and upper triangular bands L and U such that $A = LU$.

The factorization is always possible so long as none of the leading minors of the matrix is zero. The solution of the linear equations is then straightforward for, if we write them as

$$Ax = b$$

Then using $A = LU$

We have $Lux = b$

If we denote Ux by the vector y , then we obtain two sets of equations viz.,

$$Ly = b$$

And $Ux = y$

The former is solved for y by forward substitution, while the latter on subsequent back-substitution yields the required solution x .

The finite difference equations set up by part I are diagonally dominate are of the non-negative type. Further, all the equations are so normalized that the diagonal-coefficients in A are all unity.

The factors L , and U over-write A in the core store. This is always possible, since they conform to the band structure of A .

The programme assumes that A is non-symmetric. Hence, in problems which produce symmetric matrices, no advantage can be gained either in storage space or in time.

2.2.4. Part IV (the output part)

This part outputs the solution produced by part III in a presentable form.

The region described by the user in part I is now reproduced geometrically and at the nodes are written the function values. Whether the data has been read in row-wise or column-wise at the part I stage, the output here is always row-wise, which is actually the desirable form.

The two formats used to output the solution have already been explained in section 2.2.1. Further, it is not possible to output more than eight columns of the region at a time. If, however, the region has more than eight columns, as automatic page change takes place.

Stream 70 has been used for output, which allows the user the facility of obtaining the output on any device of his choice.

2.3 The working of the programme

The input is via the paper tape. The 5 procedure go after the title of the programme. The date waits till it is called. The output is through any device of the users choice.

For storing the intermediate dates, the programme requires two magnetic tapes bearing the identifiers :

DG 1112 and *DG 1113*

Part I and IV use both the magnetic tapes, while the parts II and III use the one labeled *DG 1112* only.

Part I writes on the two magnetic tapes in the following manner: on *DG 1112* are written out the band matrix and the right hand side. On *DG 1113* are written out 3 blocks of information. The first two blocks contain information subsequently necessary for arranging the output, while the third block is a vector containing . Function values at nodes on the Dirichlet boundary. As a matter of fact, this vector is also stored for use at the output time only.

As soon as part I has completed its work, part II takes over automatically. As described already , this part time the band (if necessary) and the trimmed version over-writes what was there on *DG 1112* so far.

Next the control is automatically passed on to part III, which brings the band and the right hand side into the high speed store and does the solution to the problem. The results are written back on *DG 1112* .

Finally , part IV steps up, and in accordance with the information on tape *DG 1113* outputs the results in the required format.

As the programme is progressing , from one stage to the other , indications are sent out to the user via the monitor typewriter. Thus he is able to follow the programme step by step.

We would like to conclude by emphasizing that in elliptic equations the demand is always on the store. Hence we have tried to shape the programme so that a maximum storage is left available to hold the data. The time factor is, therefore , often overlooked.

The flow chart of part I and the programme failures are appended in A(2) I and A(2)II respectively.

CHAPTER III

NUMERICAL TREATMENT OF RE-ENTRANT CORNER IN LAPLACE EQUATION IN CYLINDRICAL POLARS WITH AXIAL SYMMETRY

3.1 Introduction

A variety of problems in electrostatics, heat, fluid flow and certain fields of elasticity require the study of the partial differential equation.

$$\frac{\partial^2 \phi}{\partial p^2} + \frac{1}{p} \frac{\partial \phi}{\partial p} + \frac{1}{p^2} \frac{\partial^2 \phi}{\partial \Theta^2} + \frac{\partial^2 \phi}{\partial z^2} = 0 \quad (1)$$

in cylindrical polars (P,Q,Z) with appropriate conditions on the boundaries of the given body . Quite often, it happens that the three dimensional problem of this type has a solution which is symmetrical about an axis. In such situations the above equation reduces to the two dimensional form.

$$\frac{\partial^2 \phi}{\partial p^2} + \frac{1}{p} \frac{\partial \phi}{\partial p} + \frac{\partial^2 \phi}{\partial Z^2} \quad (2)$$

in the p-z plane. The study of equation (2) is therefore, of particular interest for axis-symmetrical problem in Mathematical physics.

In this chapter we propose to obtain the solution of equation (2) over a region with sharp corners in the boundary. No work of this nature seems to have any mention in the literature. The only references which exist so far with regard to the equations associated with Laplacian operator. The works of Motz (1946), Jeffreys and Jeffreys (1950), Woods (1953) and Wilson (1962) are noteworthy in this connection.

Equation (2) has a non-Laplacian operator, still it is possible to follow the same line of thought in treating the singularity here as employed by the above authors, in similar situations, in equations involving the Laplacian operator. However so far as working details are concerned , this agreement is not possible, obviously because the two operators are different in nature. This point has been clearly brought out in the analysis which will follow in the subsequent sections .

Further we have modified the direct methods of Motz, Jeffreys and Jeffreys and Wilson, so that now it is possible to work them iteratively. From two considerations it may be regarded as an improvement in these methods.

- (a) it eliminates the prohibitive amount of algebra in working out the special equations at certain nodes in the vicinity of the singularity.
- (b) It makes it possible to programme these methods for automatic work.

To be able to discuss the modification, it is necessary to describe the original method first. We have done so in brief outlines in section 3.3. The modification is described in section 3.5.

Lastly, in order to illustrate our approach in treating the singularity, we have set up an actual problem and propose to work it out fully.

3.2 The problem and preliminary ideas

Let us consider a circular disc of unit radius held coaxially at the middle point of a large right circular cylinder of radius $a > 1$, the axis of the cylinder being normal to the plane of the disc. If the disc carries a uniform potential V and the surface of the cylinder be earthed, the problem is to find the distribution of the potential inside the cylinder (cf. sneddon-1962a).

Due to symmetry, we can consider only one half of the cylinder. Further, since the solution is obviously symmetrical about the axis of the cylinder we need consider only half its section in the p - z plane. The three dimensional problem is thus reduced to one in two dimensions. Pictorially it is shown in fig. 1.

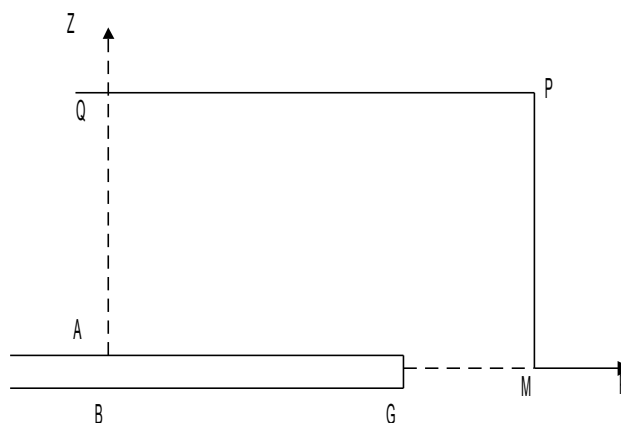


Fig.1

Here, we are required to solve (2) for the wanted function ϕ . The boundary conditions are :

1. $\phi=V$ on the disc AG ($0 \leq P \leq 1$) ;
2. $\frac{\partial \phi}{\partial Z}=0$ on CM ($1 < P < a, Z=0$) ;
3. $\phi=0$ on PM ($p=a, \text{all } Z$) ;
4. $\phi=0$ on PQ ($Z = \text{length of cylinder}$) ;
5. $\frac{\partial \phi}{\partial p}=0$ on AQ ($P=0, \text{all } Z$).

Due to sudden change in the direction of the boundary at c, a sharp corner is produced there. While the function remains finite, its derivatives tend to infinity as we approach G. Such a corner is often called the re-entrant corner. The finite difference approximation to (2) cannot take account of the behavior of the function in its vicinity. Some special treatment is therefore, necessary. As pointed out earlier (section 3.1) our method of treatment is the same as employed in equations with the Laplacian operator. It is only in the actual working details that the difference lies. We have no plane of going into the working details until we have outlined the theoretical basis of the treatment . We do so in the following section and with reference to Poisson's equation only since it is easy that way.

3.3 The Analytical Basis

As already pointed out, we will confine ourselves to poisson's equation in this section. Let the singularity at one or more boundary points be of the re-entrant corner type.

In analyzing the Risson's equation

$$\nabla^2 \phi = k \quad (4)$$

by numerical methods, the usual technique is to cover the given region by a suitable net.

If h denotes the mesh length and ϕ_0 the function value at a given node P with $\phi_1, \phi_2, \phi_3, \phi_4$ at the neighbouring ones , then we have

$$h^2 \nabla^2 \phi_0 - (\phi_1 + \phi_2 + \phi_3 + \phi_4 - 4\phi_0) = -\frac{1}{12} \left(\delta^4_x + \delta^4_y \right) \phi_0 + \dots \quad (5)$$

where δ_x^4 and δ_y^4 are fourth order central difference operators in the x and y directions respectively. If p is a node away from the singularity the right hand side of (5) converges, and a replacement of

$$h^2 \nabla^2 \phi \quad (6)$$

by

$$\phi_1 + \phi_2 + \phi_3 + \phi_4 - 4\phi_0 \quad (7)$$

is valid, the local truncation error is bounded and is of the order h^4 .

On the other hand, if P is close to a singularity the right hand side (5) diverges. Accordingly, a replacement of (6) by (7) at such a node will give unsatisfactory solution in a field local to P. Hence the necessity of some special treatment of such nodes.

The general method to do this is to obtain a series solution to (4) satisfying the given boundary conditions near the singularity and then fit it appropriately in the vicinity of the singularity.

For example, let us refer to the re-entrant corner as shown in fig.2. The boundary condition could be (i) either, function value ϕ prescribed in the boundary, or (ii) the normal derivative $\frac{\partial \phi}{\partial n}$ prescribed on the boundary.

Let us first consider the case when the boundary condition of the type (i) is prescribed. Hence let $\phi = \phi_0$ on the two arms of the boundary, and let ϕ remain finite as we approach 0. Then the appropriate series.

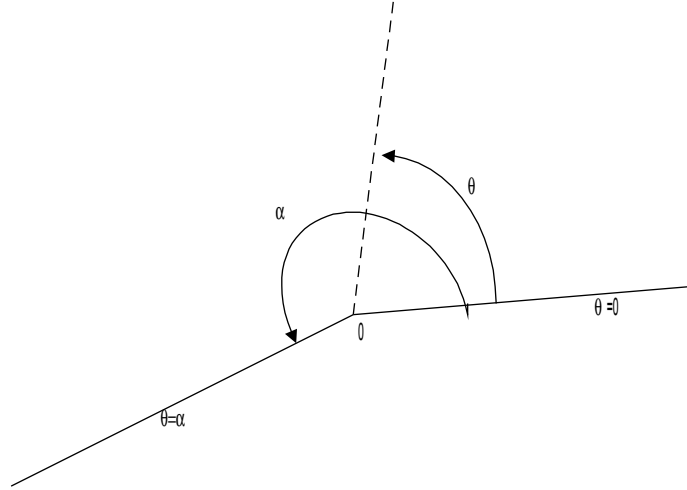


Figure 2.

Solutions, satisfying the boundary condition is

$$\phi = \phi_0 + \sum_{n=1}^{\infty} a_n r^{\frac{n\pi}{\alpha}} \sin \frac{n\pi}{\alpha} \Theta \quad (8)$$

Next, in the case of (II) - type boundary condition, if the normal derivative vanishes on the boundaries joining at 0, then the solution is

$$\phi = \sum_{n=0}^{\infty} b_n r^{\frac{n\pi}{\alpha}} \cos \frac{n\pi}{\alpha} \Theta \quad (9)$$

(Here the coordinates r, Θ signify that the given region has been considered in plane polar context).

It is clear from both (8) and (9) that when $\alpha > \pi$ the index of $r < 1$ and, therefore $\frac{\partial \phi}{\partial r} \rightarrow \infty$ as $r \rightarrow 0$, Thus the series (8) and (9) have the relevant properties - they are finite with derivatives discontinuous at $r \rightarrow 0$.

Thus having obtained the appropriate series solution (8) or (9), the next step is to fit it in the neighbourhood of the singularity so that continuity in the solution is obtained with the rest of the region where a standard finite difference formula has been used.

To meet this end, the coefficients a's and b's in the two series have to be evaluated in a suitable manner.

It is at this juncture that Motz, Jeffreys and Jeffreys, Woods and Wilson depart and put forward their own methods for evaluating the coefficients a's and b's. Of course, Woods uses the same series (8)/(9) but his approach is slightly different. At the moment we confine ourselves to the methods of Motz, Jeffreys and Wilson and describe them briefly with reference to an example.

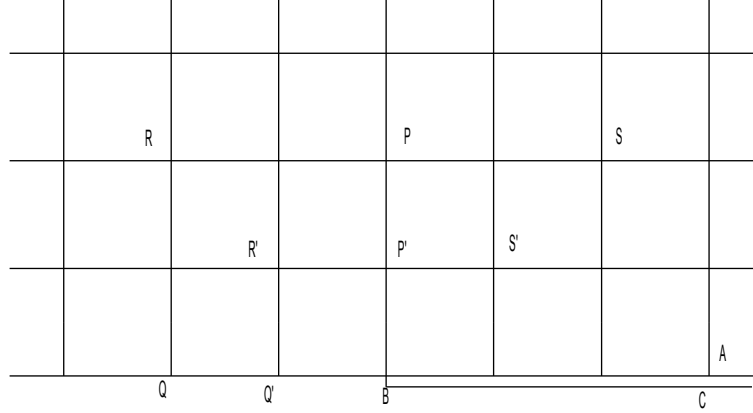


Fig.3

In fig. 3, let B be the re-entrant corner, the two arms of the boundary being AB and BC, so that $\alpha = 2\pi$ here. Also let $\phi = \phi_0$ on ABC and $\frac{\partial \phi}{\partial \nu} = 0$ on BD. The rest of the region is regular and closed with suitable conditions prescribed on the boundary. Of course, the wanted function satisfies Poisson's equation.

Motz-method requires a finite number of terms of (8) to be taken and therefore, as many nodes taken in the neighbourhood of B. He calls these nodes as special points. For instance, if two terms of (8) are taken, i.e.

$$\phi = \phi_0 + a_1 r^{\frac{1}{2}} \sin \frac{\Theta}{2} + a_2 r \sin \Theta \quad (10)$$

then the two special points could be the ones marked P' and Q'. Assuming that the solution (10) is also true at P and Q, which are a step further away, it follows that

$$\phi = \phi_0 + a_1 r_P^{\frac{1}{2}} \sin \frac{1}{2} \Theta_P + a_2 r_P \sin \Theta_P$$

and (11)

$$\phi_Q = \phi_0 + a_1 r_Q^{\frac{1}{2}} \sin \frac{1}{2} \Theta_Q + a_2 r_Q \sin \Theta_Q$$

Equations (11) are linear simultaneous equations in two unknowns a_1 and a_2 . Hence on solving them a_1 and a_2 are determined as linear combination of ϕ_P and ϕ_Q . They could be written as

$$\begin{aligned} a_1 &= f_1(\phi_P, \phi_Q) \\ \text{and} \quad a_2 &= f_2(\phi_P, \phi_Q) \end{aligned} \quad (12)$$

On substituting these values in (10), two special equations are obtained for the special points P' and Q', viz.

$$\begin{aligned} \phi_{P'} &= F_1(\phi_P, \phi_Q) \\ \text{and} \quad \phi_{Q'} &= F_2(\phi_P, \phi_Q) \end{aligned} \quad (13)$$

Thus using the special equation at special point P' and Q' and the standard finite difference equations at all other nodes, including P and Q, there are as many equations as the number of unknowns. The solution is then obtained in one go if an electronic computer is available.

In the determination of the coefficients a_1 and a_2 by the method of Jeffreys and Jeffreys the influence of all the neighbouring points is taken into account. Thus if we consider P', Q', R', S' as the special points this time, we have four equations of the type of (11) in place of two. Using the theory of least squares the "best" estimates of a_1 and a_2 are obtained from them. Again, we have four special points and the standard finite difference equations at the rest of the nodes, we have, again as many equations as there are unknowns.

By taking into account the influence of all the neighbouring points, Jeffrey and Jeffreys have tried to improve upon Motz's coefficients. But the algebra involved in it is certainly prohibitive.

Wilson, however, thinks in a different way. Since the field changes rapidly in the vicinity of the singularity, he suggests that each special point should have its own set of coefficients. Considering only the 'dominant' term in (8) he writes;

$$\phi = \phi_0 + a_1 r^{\frac{1}{2}} \sin \frac{1}{2} \Theta + [px + qy + s(x^2 - y^2) + txy + v] \quad (14)$$

in place of (10) above. Here, the terms in square brackets are a Taylor series expansion about the node at which the special equation is to be computed. v, p, q, a and t are additional arbitrary constants to be determined. At a special point and its neighbouring nodes, he sets up linear equations like equation (14), given them special weighting, and solves them using “least squares” technique to obtain the “best” estimates of the arbitrary constants involved in them. He applies this process to each of the special points and thus obtains a separate set of coefficients for each one of them.

Thus having described the analysis and the different methods for treating the singularity, we now summarise, the general terms, all the ideas we encountered in this section.

3.4 Summary

Broadly speaking the whole process of treating the singularity involves two steps :

Step 1 To obtain an appropriate series solution to the given differential equation, satisfying
the boundary conditions at the singularity.

Step 2 To obtain the coefficients a 's, b 's of the series solution such that while a “good fit” of
this solution has been obtained in the neighbourhood of the singularity, it is continuous
with the solution in the rest of the region.

3.5 Modification of the Direct Methods

We are now in a position to describe how the modification of the method of Motz, Jeffreys and Wilson has been devised. We will do so with reference to Motz's method and the example considered in section 3.3 above.

Let us again assume P' and Q' to be the special points (fig. 3). Accordingly, we have the relation (10), again consisting of two terms of the appropriate series expansion.

Now let $\phi_P^{(0)}$ and $\phi_Q^{(0)}$ be the trial values of ϕ at the further away nodes P and Q, then in place of (11) we write

$$\begin{aligned}\phi_P^{(0)} &= \phi_0 a_1 r_P^{\frac{1}{2}} \sin \frac{1}{2} \Theta_P + a_2 r_P \sin \Theta_P \\ \text{and} \\ \phi_Q^{(0)} &= \phi_0 a_1 r_Q^{\frac{1}{2}} \sin \frac{1}{2} \Theta_Q + a_2 r_Q \sin \Theta_Q\end{aligned}\tag{A}$$

again these are linear simultaneous equations, but this time the algebraic quantities involved in them are only a_1 and a_2 . Hence, on solving them we obtain the first estimates of the coefficients a_1 and a_2 . Let us call them $a_1^{(0)}$ and $a_2^{(0)}$ respectively. Now, corresponding to them the function values at P' and Q' from (10) are

$$\begin{aligned}\phi_{P'}^{(1)} &= \phi_0 a_1^{(1)} r_{P'}^{\frac{1}{2}} \sin \frac{1}{2} \Theta_{P'} + a_2^{(1)} r_{P'} \sin \Theta_{P'} \\ \text{and} \\ \phi_{Q'}^{(1)} &= \phi_0 a_1^{(1)} r_{Q'}^{\frac{1}{2}} \sin \frac{1}{2} \Theta_{Q'} + a_2^{(1)} r_{Q'} \sin \Theta_{Q'}\end{aligned}\tag{B}$$

Relations (B) are no more algebraic special equation. Their right hand side being completely known they are the numerical estimates of the function values at the special points P' and Q'. Momentarily we fix these estimates at P' and Q', and relax the rest of the region using a standard finite difference formula. This gives us fresh estimates $\phi_P^{(1)}$ and $\phi_Q^{(1)}$ at P and Q respectively. We repeat the cycle over and over again till finally the process converges. As the only bit of algebra involved here is the solution of the linear simultaneous equations (A) it is easy to programme the whole iteration procedure for automatic computation.

It is easy to see that there can be no difficulty in likewise modifying the methods of Jeffreys and Jeffreys and Wilson.

We have tested the modified versions of Motz's and Jeffreys and Jeffreys methods on the examples solved by them in their respective works. The results were satisfactory.

3.5 Back to our problem

As this stage we return to our problem already stated in section 3.2, and propose to treat the re-entrant corner occurring in (2) in exactly the same manner as outlined in steps 1 and 2 mentioned in section 3.4.

Hence, as step 1 we proceed to find out the series solution to (2), satisfying the boundary conditions that $\phi=V$ on $AG(\Theta=\pi)$ and $GB(\Theta=\pi)$.

We transfer the origin to G by using the transformation:

$$s = P - 1$$

We thus obtain from (2)

$$\frac{\partial^2 \phi}{\partial s^2} + \frac{1}{1+s} + \frac{\partial^2 \phi}{\partial z^2}$$

On changing this into plane polar coordinates, we get on using

$$r^2 = a^2 + z^2 \quad \text{and} \quad \tan \Theta = \frac{z}{s}$$

the equivalent form

$$\begin{aligned} & \frac{\partial^2 \phi}{\partial r^2} + \frac{\partial \phi}{\partial r} + \frac{1}{r^2} \frac{\partial^2 \phi}{\partial \Theta^2} + \frac{1}{1+r \cos \Theta} \left[\frac{\partial \phi}{\partial r} \cdot \frac{\partial r}{\partial s} + \frac{\partial \phi}{\partial \Theta} \cdot \frac{\partial \Theta}{\partial s} \right] = 0 \\ \text{or } & \frac{\partial^2 \phi}{\partial r^2} + \frac{1}{r} \frac{\partial \phi}{\partial v} + \frac{1}{v^2} \frac{\partial^2 \phi}{\partial \Theta^2} + (1+r \cos \phi)^{-1} \left[\frac{\partial \phi}{\partial v} \cos \Theta - \frac{\sin \Theta}{v} \frac{\partial \phi}{\partial \Theta} \right] = 0 \end{aligned} \quad (15)$$

Let the solution to (15) be

$$\phi = r^2 [f_0(\Theta) + r f_1(\Theta) + r^2 f_2(\Theta) + \dots + r^n f_n(\Theta) + \dots] \quad (16)$$

$$\text{then } \frac{\partial \phi}{\partial r} = \alpha^{\alpha-1} f_0 + (\alpha+1)r^\alpha f_1 + (\alpha+2)r^{\alpha+1} f_2 + \dots + (\alpha+n)r^{\alpha+n-1} f_n + \dots$$

$$\frac{\partial^2 \phi}{\partial r^2} = \alpha(\alpha-1)r^{\alpha-1} f_0 + (\alpha+1)\alpha^{\alpha-1} f_1 + (\alpha+2)(\alpha+1)r^\alpha f_2 + \dots + (\alpha+n)(\alpha+n-1)r^{\alpha+n-2} f_n + \dots$$

$$\frac{\partial \phi}{\partial \Theta} = r^\alpha f_0' + r^{\alpha+1} f_1' + \dots + r^{\alpha+n} f_n' + \dots$$

$$\frac{\partial^2 \phi}{\partial \Theta^2} = r^\alpha f_0'' + r^{\alpha+1} f_1'' + \dots + r^{\alpha+n} f_n'' + \dots$$

On substituting these values in (15) we get

$$\sum_{n=0} (\alpha+n)(\alpha+n-1)r^{\alpha+n-2} f_n + \sum_{n=0} r^{\alpha+n-2} f_n'' + (1-r \cos \Theta + r^2 \cos^2 \Theta - r^3 \cos^3 \Theta + \dots)$$

$$\left[\sum_{n=0} (\alpha+n)r^{\alpha+n-1} f_n \cos \Theta - \sum_{n=0} r^{\alpha+n-1} f_n' \right] \sin \Theta = 0$$

On rearrangement we get

$$(f_0'' + \alpha^2 f_0) r^{\alpha-2} + [f_1'' + (\alpha+1)^2 f_1 + \alpha f_0 \cos \Theta - f_0' \sin \Theta] r^{\alpha-1}$$

$$[f_2'' + (\alpha+2)^2 f_2 + (\alpha+1) f_1 \cos \Theta - \alpha f_0 \cos^2 \Theta - f_1' \sin \Theta + f_0' \sin \Theta \cos \Theta] r^{\alpha}$$

$$[f_3'' + (\alpha+3)^2 f_3 + (\alpha+2) f_2 \cos \Theta - (\alpha+1) f_1 \cos^2 \Theta + \alpha f_0 \cos^3 \Theta - f_2' \sin \Theta + f_1' \sin \Theta \cos \Theta - f_0' \sin \Theta \cos^2 \Theta] r^{\alpha+1} + \dots = 0$$

Since this power series is identically equal to zero, all its co-efficient must vanish separately. Hence

$$f_0'' + \alpha^2 f_0 = 0$$

$$\text{or } f_0 = a_0 \sin \alpha \Theta + b_0 \cos \alpha \Theta$$

Next ,

$$f_1'' + (\alpha+1)^2 f_1 + \alpha f_0 \cos \Theta - f_0' \sin \Theta = 0$$

$$\text{Which gives } f_1 = a_1 \sin(\alpha+1)\Theta + b_1 \cos(\alpha+1)\Theta - \frac{1}{4} [a_0 \sin(\alpha-1)\Theta + b_0 \cos(\alpha-1)\Theta]$$

$$\text{Similarly, } f_2 = a_2 \sin(\alpha+2)\Theta + b_2 \cos(\alpha+2)\Theta - \frac{4a_1(\alpha+1) - a_0(1+2\alpha)}{16(\alpha+1)} \sin \alpha \Theta \\ - \frac{4b_1(\alpha+1) - b_0(1+2\alpha)}{16(\alpha+1)} \cos \alpha \Theta + \frac{3}{32} [a_0 \sin(\alpha-2)\Theta + b_0 \cos(\alpha-2)\Theta],$$

And so on.

In our particular problem, the series solution should be such that it vanishes on $AG(\Theta=\pi)$ and $GB(\Theta=\pi)$. Hence we have $\alpha = \frac{1}{2}$ and all $a's = 0$.

We can also assume that as many as only two terms of the series expansion would be sufficient for our purposes.

Hence we neglect terms with coefficients $b_n \gg 1$. In this way we are able to simplify the expression for f_0, f_1, f_2 etc.

$$f_0 = b_0 \cos \frac{\Theta}{2}$$

$$f_1 = -\frac{1}{4} b_0 \cos \frac{1}{2} \Theta$$

$$f_2 = \left(\frac{1}{12} \cos \frac{1}{2} \Theta + \frac{3}{32} \cos \frac{3}{2} \Theta \right) b_0 - \frac{1}{4} b_1 \cos \frac{1}{2} \Theta$$

$$f_3 = -\left(\frac{1}{16} \cos \frac{\Theta}{2} + \frac{1}{30} \cos \frac{3\Theta}{2} + \frac{5}{128} \cos \frac{5\Theta}{2} \right) b_0 + \left(\frac{3}{32} \cos \frac{\Theta}{2} + \frac{1}{10} \cos \frac{3\Theta}{2} \right) b_1$$

etc.

Finally on substituting these and several more expressions like this in (16) and remembering that $\phi = V$ on the disc AG, we write

$$\begin{aligned} \phi = V + b_0 & \left[r^{\frac{1}{2}} \left(1 - \frac{r}{4} + \frac{r^2}{12} - \frac{r^3}{16} + \frac{3r^4}{80} - \frac{r^5}{32} + \frac{5r^6}{224} - \frac{5r^7}{256} + \dots \right) \cos \frac{\Theta}{2} + r^{\frac{5}{2}} \left(\frac{3}{32} - \frac{r}{30} + 5r^2 - 3r^3 + 35r^4 - 5r^5 + \dots \right) \right. \\ & - r^{\frac{7}{2}} \left(\frac{5}{128} - \frac{r}{70} + \frac{35r^2}{1536} - \frac{r^3}{84} + \frac{63r^4}{4096} \dots \right) \cos \frac{5\Theta}{2} + r^{\frac{9}{2}} \left(\frac{35}{2048} - \frac{2r}{315} + \frac{105r^2}{8192} - \frac{r^3}{154} + \dots \right) \cos \frac{7\Theta}{2} \\ & \left. - r^{\frac{11}{2}} \left(\frac{63}{8192} - \frac{2r}{693} + \frac{99r^2}{8192} - \dots \right) \cos \frac{9\Theta}{2} + r^{\frac{13}{2}} \left(\frac{231}{65536} - \frac{4r}{3003} \right) \cos \frac{11\Theta}{2} - \frac{3003r^{\frac{15}{2}}}{1048576} \cos \frac{13\Theta}{2} + \dots \right] \\ & + b_1 \left[r^{\frac{3}{2}} \left(1 + \frac{r^2 10}{128} - \frac{5r^3}{560} + \frac{27r^4}{256} - \frac{7r^5}{168} + \frac{5r^6}{168} - \dots \right) \cos \frac{3\Theta}{2} - r^{\frac{5}{2}} \left(\frac{1}{4} - \frac{3r}{32} + \frac{3r^2}{40} - \frac{3r^3}{64} + \frac{9r^4}{224} - \frac{15r^5}{512} + \dots \right) \cos \frac{\Theta}{2} \right. \\ & - r^{\frac{7}{2}} \left(\frac{3}{70} - \frac{35}{2048} + \frac{r^2}{35} - \frac{63r^2}{4096} + \dots \right) \cos \frac{5\Theta}{2} + r^{\frac{9}{2}} \left(\frac{2}{105} - \frac{63r}{8192} + \frac{5r^2}{308} - \dots \right) \cos \frac{7\Theta}{2} \\ & \left. - r^{\frac{11}{2}} \left(\frac{2}{231} - \frac{231r}{65536} + \dots \right) \cos \frac{9\Theta}{2} + \frac{4r^{\frac{13}{2}}}{1001} \cos \frac{11\Theta}{2} - \dots \right] \end{aligned} \quad (17)$$

In view of the slow convergence of the series expansion, we went as far as f_7 .

Thus having obtained the series expansion, we have finished with step 1. We now proceed ahead to step 2. Here, we have to obtain suitable estimates of the coefficients b_0 and b_1 occurring in the series expansion (17). We have employed the modified versions of the Motz's and the

Jeffreys and Jeffrey's methods towards this end . They have already been described in section 3.5. Hence we now proceed to outline the numerical details and the sets of results we have obtained.

3.7. The numerical details

Let the radius AM of the cylinder be $\frac{5}{3}$ units and its semi length $AQ=5$ units (fig. 1). Further, let us cover the region AGMPQ by a square net of mesh length = $\frac{1}{6}$. Now , if the potential on the disc be 1000, then we set $V=1000$ in (17).

In figure 4 we have drawn a small portion of the region near the singularity.

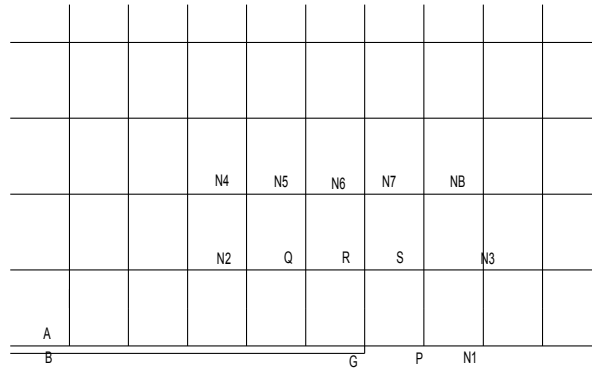


Figure 4.

With reference to this figure we would point out the special points and the further away nodes considered for the different results . In all , nine sets of results, have been worked out – eight by the modified Jeffreys and Jeffreys method (briefly called m-j.j) and only one by the modified Motz's method (called m-M). The details are as shown on the next page.

Table 1

No.	Meth od	Special Points	Further away nodes	Coefficie nt b_0	Coefficie nt b_1	Number of iteration s
1	m-J.J.	P, Q, R, S	N1 N4 N6 N8	- 1415042	-	8
2	m-J.J.	P, Q, R, S	N1 N4 N6 N8	-1465.42	42.60	10
3	m-J.J.	P, Q, R, S	N1 N2 N3 N5 N6 N7	-1421.27	-	10
4	m-J.J.	P, Q, R, S	N1 N2 N3 N5 N6	-1400.85	58.04	10

			N7			
5	m-J.J.	P, Q, R, S	N1 TO N8	-1418.74	-	9
6	m-J.J.	P, Q, R, S	N1 TO N8	-1432.60	32.93	10
7	m-J.J.	P, Q, R, S	N1 TO N9	1431.74	-	9
8	m-J.J.	P, Q, R, S	N1 TO N9	1440.16	36.99	9
9	m-M	P, R	N1, N6	1433.66	117.55	11

Symmetries about the axis of the cylinder and the disc were taken into account in all the versions of the method of Jeffreys and Jeffreys. The equation at N1 was weighted twice while at rest of the nodes they were weighted four times.

The actual results, from all the nine cases above, have been recorded in appendix A(3)I to A(3)IX. For ready convenience, however, we have in fig. 5, a magnified picture of the region near the singularity. At each node are the function values as obtained in the different cases. These cases have already been serially ordered in table I, and the function values from them have been arranged in the same order for example, the function values at the top correspond to the case marked I in table I.

Table I clearly indicates the special points considered in each version of the methods listed there. They are called special points because special equations are used at them. In contrast, at rest of the nodes, ordinary finite difference equations are used. The five-point finite difference representation to (2) which we have used here is

$$\left(1 + \left(\frac{h}{2p_0} \right) \phi_1 + \phi_2 + \left(1 - \frac{h}{2p_0} \right) \phi_3 + \phi_4 - 4\phi_0 = 0 \right) \quad (18)$$

where the subscripts have their usual meaning.

307 302 307 307 307 305 306 304 295	265 261 265 266 265 264 263 255	216 211 215 216 215 214 214 213 208	161 157 160 161 161 160 160 159 155	105 102 105 105 105 104 104 104 101
382 387 382 382 382 380 381 389 376	340 334 340 341 340 338 339 337 327	276 270 275 277 276 274 274 273 266	204 199 204 205 204 202 203 202 198	132 128 132 133 132 131 131 130 128
504 487 503 504 504 501 502 500 480	439 431 438 439 439 436 437 435 421	355 347 354 356 354 352 352 350 342	259 251 258 261 258 256 256 255 251	165 160 164 166 164 163 163 162 160
656 648 655 655 655 652 654 651 614	572 562 572 572 572 568 570 567 548	480 449 459 461 459 456 457 454 446	325 314 324 329 324 321 321 319 317	201 194 200 204 200 199 198 197 196
808 802 808 807 808 805 807 804 787	755 742 754 751 754 748 752 747 728	608 582 607 610 608 602 604 600 587	399 379 396 407 397 393 392 390 381	235 225 234 239 234 232 231 230 231
		307 302 307 307 307 305 306 304 285	444 427 441 453 442 439 437 436 444	252 241 250 254 251 248 248 247 249

Figure 5.


$$\left[\frac{h^4}{P_0 \angle 3} \frac{\partial^3 \phi_0}{\partial P_0^3} + \frac{h^4}{12} \left(\frac{\partial^4 \phi_0}{\partial p_0^4} + \frac{\partial^4 \phi_0}{\partial Z_0^4} \right) \right] + \frac{h^5}{P_0 \angle 5} \frac{\partial^5 \phi_0}{\partial P_0^5} + \dots (19)$$

Using ϕ as given by the series solution (17), we can obtain an estimate of this error at different nodes in each version of the above methods. The first three terms of (19), viz. those, which are in square brackets, are the principal contributors to this error.

Fig. 6 shows its magnitude, in terms of the coefficients b_0 and b_1 , at some of the nodes in the vicinity of the singularity. On substituting the values of b_1 and b_1 from table I, we obtain the numerical estimate of this error at each node. It is shown in fig. 7. The error, corresponding to each method, has been arranged at a node in the same serial order as the method itself in table I.

It is interesting to find from fig.7 that in all the nine cases, the value of this error at each node is almost the same. It may indicate that in order to take account of the singularity it does not really matter whether we consider the series solution (17) as far as term in b_1 or just in b_0 .

The four nodes P, Q, R and S are obviously, the dangerous ones. As was expected, the errors are large at them all.

In Motz's method (case 9, table I), only two of these four nodes have been dealt with rather squarely. They are P and R (fig. 7), where special equations have been used. However, at Q and S, where the finite difference equations have been used, the solution is bound to be more inaccurate, and the local neighbourhood will also be affected, appropriately.

In all the versions of the method of Jeffreys and Jeffreys (cases 1 to 8, table I), special equations have been used at all the four nodes P, Q, R and S. In this sense, this method can be regarded to promise better results than Motz's method. In theorem I, section 90, Milne (1957) has given an expression for estimating the upper bound of the error in the finite difference solution of the Laplace's equation. If we choose to use the same expression here as well, then it appears that the solutions obtained from the method of Jeffreys and Jeffreys have a maximum error of about 13 units.

In appendix A(3) X is the solution obtained by ignoring the presence of singularity. It is included here for purposes of comparison.

Concluding Remarks

We could not try Wilson's method here. In addition to the series expansion about the corner, it also required a Taylor series about the node. It is a limitation with this method.

So far we did not make any mention of Wood's method. However, we did try his method over our problem, but found that it did not converge. With Laplacian operator it converges, even under such situations as we had here. This behavior of the method needs investigation. We have not attempted to explore it.

CHAPTER IV

EXTENTION OF MILNE'S METHOD TO BIHARMONIC EQUATION

4.1. Introduction

Eigenvalue problems arise in a variety of ways in Mathematical Physics, and whether they appear associated with the ordinary or the partial differential equations, the finite difference procedure of their solution is to reduce them to a matrix of the form

$$(A - \lambda_i I)x_i = 0 \quad (1)$$

and then calculate the latent roots λ_i and vectors x_i of the matrix A.

This sort of reduction of a continuous problem to one of discrete nature is the basis of the finite difference theory. The finite difference approximation need be reasonably accurate therefore, otherwise, the ill effects of discretisation are sometimes evident even in the least latent roots.

Milne (1957) has proposed a method whereby, using not too refined relaxation patterns for the different differential operators, it is possible to obtain results which would be better than those which one would ordinarily obtain from (1). He has applied it to the eigenvalue problems in the

ordinary differential equations and the partial differential equations with the two dimensional Laplacian operator. In this chapter the same technique has been extended to the partial differential equations associated with the bi-harmonic operator

$$\nabla^4 \equiv \frac{\partial^4}{\partial x^4} + 2 \frac{\partial^4}{\partial x^2 \partial y^2} + \frac{\partial^4}{\partial y^4}$$

in two dimensions.

4.2. The Analysis

Associated with appropriate boundary conditions , the general equation governing the free vibrations in a plate is

$$\nabla^4 u = P_i^4 u \quad (2)$$

where $u = u(x, y)$ is the transverse displacement of the plate, held in the x-y plane; and p_i , called the eigenvalues, are purely numerical quantities such that a non-trivial solution of (2) is valid. The problem is to determine the eigenvalue P_i .

Let us consider an arbitrary function $w = w(x, y, t)$ satisfying the parabolic equation

$$\frac{\partial w}{\partial t} = c^2 \nabla^2 w \quad (3)$$

Differentiating (3) we get the sequence

$$\frac{\partial^2 w}{\partial t^2} = c^4 \nabla^4 w \quad (4)$$

$$\frac{\partial^3 w}{\partial t^3} = c^6 \nabla^6 w \quad (5)$$

Let us cover the plate with a square mesh of width h and let k be the direction of t. Then by Taylor's Theorem, we have

$$\begin{aligned}
w(t+2k) - 2w(t+k) + w(t) &= k^2 \frac{\partial^2 w}{\partial t^2} + k^3 \frac{\partial^3 w}{\partial t^3} + \frac{7}{12} k^4 \frac{\partial^4 w}{\partial t^4} + \dots \\
&= k^2 c^4 \nabla^4 w + k^3 c^6 \nabla^6 w + \frac{7}{12} k^4 c^8 \nabla^8 w + \dots \\
&= \frac{1}{36} h^4 \nabla^4 w + \frac{1}{216} h^6 \nabla^6 w + \frac{7}{15552} h^8 \nabla^8 w + \dots
\end{aligned}$$

(6)
where

$$6kc^2 = h^2 \quad (7)$$

Substituting it in (6) and neglecting higher order terms in h we get

$$\begin{aligned}
\left(\ell^{-2c^2 P^2 k} - 2\ell^{-c^2 P^2 k} + 1 \right) u &= \frac{1}{36} h^4 \nabla^4 u \\
\text{or} \quad h^4 \nabla^4 u &= 36 \left(1 - \ell^{-\frac{h^2 P^2}{6}} \right)^2 u
\end{aligned}$$

Employing a suitable relaxation pattern for $h^4 \nabla^4$ we obtain a matrix for the region of the plate. Let λ_i be the latent roots of this matrix, then

$$\begin{aligned}
\lambda_i &= 36 \left(1 - \ell^{-\frac{h^2 P_i^2}{6}} \right)^2 \\
\text{or } h^2 P_i^2 &= -6m \left(1 - \frac{\sqrt{\lambda_i}}{6} \right) \quad (8)
\end{aligned}$$

which gives the required eigenvalues P_i

4.3 **Applications of the formula to rectangular plates**

A coarse relaxation pattern for $h^4 \nabla^4$ as

$$\begin{array}{ccccc}
& & 1 & & \\
& 2 & -8 & 2 & \\
1 & -8 & +20 & -8 & 1 \\
& 2 & -8 & 2 & \\
& & 1 & &
\end{array}$$

and for a rectangular plate, supported at its four edges, λ is given by

$$4 \left[2 - \cos \frac{m\pi h}{\alpha} - \cos \frac{n\pi h}{\beta} \right] \quad (9)$$

Where α and β the length and breadth of the plate and m, n are integers. Substituting it in (8) we get

$$h^2 P^2 = 6m \left[1 - \frac{2}{3} \left(\sin^2 \frac{m\pi h}{2\alpha} + \sin^2 \frac{n\pi h}{2\beta} \right) \right] \quad (10)$$

$$\text{or } P^2 = \pi^2 \left(\frac{m^2}{\alpha^2} + \frac{n^2}{\beta^2} \right) + \frac{\pi^4 m^2 n^2 h^2}{6\alpha^2 \beta^2} + \frac{\alpha^6 h^4}{108} \left[\frac{2m^6}{\alpha^6} + \frac{m^4 n^2}{\alpha^4 \beta^2} + \frac{m^2 n^4}{\alpha^2 \beta^4} + \frac{2n^6}{\beta^6} \right] + \dots \quad (11)$$

The similar expression as would be obtained direct from (9) is

$$\pi^2 \left(\frac{m^2}{\alpha^2} + \frac{n^2}{\beta^2} \right) - \frac{\pi^4 h^2}{12} \left(\frac{m^4}{\alpha^4} + \frac{n^4}{\beta^4} \right) + \frac{\pi^6 h^4}{36} \left(\frac{m^6}{\alpha^6} + \frac{n^6}{\beta^6} \right) + \dots \quad (12)$$

The first term in the series (11) and (12) are the true values of P^2 , the remaining terms being the error terms in both the series.

The remaining terms being the error terms in both the series are, obviously, the dominant ones, and it is easy to see that the first error terms in (11) are smaller than their counterparts in (12) except when $m = n$ in a square plate. Hence it is clear that (11) will give a better estimate of P^2 as obtained from (11) and (12) for a 2 x 3 rectangular plate :

No	True value of P^2	H = 1/4		H = 1/8		H=1/12	
		From (11)	From(12)	From(11)	From (12)	From (11)	From (12)
1	3.564	3.593	3.526	3.571	3.554	3.565	3.563
2	6.854	6.970	6.723	6.882	6.821	6.857	6.850
3	10.966	11.078	10.994	10.994	10.838	10.969	10.951
4	12.337	12.600	12.401	12.401	12.203	12.344	12.322
5	14.256	14.725	14.370	14.370	14.105	14.269	14.239

While the true value were computed by $P^2 = \frac{\pi^2}{36}(4m^2 + 9n^2)$

The approximate values were calculated from (9) and (10) not from the series (11) and (12). It is easy to see that first- column-approximate- values for each h are closer to the true values than the second-column- approximate values.

So far we employed a poor relaxation pattern for $h^4 \nabla^4$. However, using a better relaxation pattern we can expect highly improved results. One such relaxation pattern is

$$3h^4 \nabla^4 = \begin{array}{ccccc} & & 1 & & 1 & & 1 \\ & & 1 & -2 & -10 & -2 & 1 \\ & 1 & -10 & 36 & -10 & 1 & \\ & 1 & -2 & -10 & -2 & 1 & \\ & & 1 & & 1 & & 1 \end{array} o(h^6)$$

we have taken it from Collatz (1959) with this relaxation pattern for $h^4 \nabla^4$ and time interval defined by (7) we can approximate (4) to a high order of accuracy - the error involved is $o(h^8)$.*

Using this relaxation pattern and taking $h = \frac{1}{4}$ we again solve the above problem. The least latent root of the resulting matrix is found to be 0.0478093 by Lanozos's method. We substitute this value in formula (8).

$$\begin{aligned} h^2 P^2 &= -6 \ln \left(1 - \frac{\sqrt{.0478093}}{6} \right) \\ &= -6 \ln (.96355778) \\ &= +6 \times (.0371228) \\ &= .2227368 \\ P^2 &= 3.56379 \end{aligned}$$

This is an agreeable estimate.

If we choose not to use formula (8) , then a value of the same order Of accuracy and with $h = \frac{1}{4}$ could be found out by Lanozos's iterative

* Let us denote the operator

$$\begin{array}{ccccc} & & 1 & & 1 & & 1 \\ & & 1 & -2 & -10 & -2 & 1 \\ & 1 & -10 & 36 & -10 & 1 & \\ & 1 & -2 & -10 & -2 & 1 & \\ & & 1 & & 1 & & 1 \end{array}$$

By p. Then using Bickley's notation (1948) we can write

$$\begin{aligned}
 (P=36) &= -10S_1 - 2S_2 + S_3 + S_4 \\
 &= -40 - 10h^2\nabla^2 - \frac{20}{L4}h^4(\nabla^4 - 2D^4) - \frac{20}{L6}h^6(\nabla^6 - 3D^4\nabla^2) - \frac{20}{18}h^8(\nabla^8 - 4D^4\nabla^4 + 2D^8) - \dots \\
 &\quad - 8 - 4h^2\nabla^2 - \frac{8}{L4}h^4(\nabla^4 + 4D^4) - \frac{8}{L6}h^6(\nabla^6 + 12D^4\nabla^2) - \frac{8}{16}h^8(\nabla^8 + 24D^4\nabla^4 + 6D^8) + \dots \\
 &\quad + 8 + 10h^2\nabla^2 + \frac{4}{L4}h^4(17\nabla^4 + 14D^4) + \frac{4}{L6}h^6(65\nabla^6 + 105D^4\nabla^2) + \frac{4}{L8}h^8(257\nabla^8 + 876D^4\nabla^4 - 1054D^8) - \dots \\
 &= -36 + 3h^4\nabla^4 + \frac{1}{2}h^6\nabla^6 + 0(h^8) \\
 3h^4\nabla^4 &= P - \frac{1}{2}h^6\nabla^6 + 0(h^8)
 \end{aligned}$$

method only by employing a more complicated relaxation pattern. Thus taking

$$6h^4\nabla^4 = \begin{array}{ccccccc} & & & 1 & & & \\ & & & 14 & & & \\ & & 1 & 20 & -77 & 20 & 1 \\ & 1 & 14 & -77 & 184 & -77 & 14 & 1 \\ & & 1 & 20 & -77 & 20 & 1 & \\ & & & 1 & 14 & 1 & & \\ & & & & 1 & & & \end{array} + 0(h^8)$$

the value found iteratively is

$$P^2 = 3.5631296$$

It is easy to verify that the value found above from formula (8) is better than this , since the true value is

$$P^2 = 3.5640238$$

We thus notice that using Milne's technique we can expect better promise from a suitable relaxation pattern of just reasonable accuracy than from a relaxation pattern of higher accuracy without it.

Substituting it in (4) we have

$$w = (2k + t) - 2w(t + k) + w(t) = \frac{k^2 c^4}{3h^4} \left[P - \frac{1}{2} h^4 \nabla^6 + O(h^8) \right] w + k^3 c^6 \nabla^6 w + \frac{7}{12} k^4 c^8 \nabla^8 w$$

$$= \frac{k^2 c^4}{3h^4} Pw - \frac{k^2 c^4}{6} (h^2 - 6kc^2) \nabla^6 w + O(h^8)$$

$$= \frac{k^2 c^4}{3h^4} Pw + O(h^8)$$

on using (7).

CHAPTER V

NUMERICAL SOLUTION OF TWO FREDHOLM TYPE INTEGRAL EQUATIONS OF SECOND KIND

5.1. Introduction

Most problem in Mathematical physics are stated in the form of differential equations and the standard methods of their solution present an interesting field of study. While the method of solving the given differential equation directly, by analytical or numerical methods, still continues to hold its well-established place, another method has been developed during the recent decades. It calls for transforming the given differential equation into a suitable integral equation whose solution is then obtained by affecting the integration involved in it. The boundary conditions are implicitly inherent in the integral equation and are not expressed separately as in a differential equation. Many boundary-value problems in Mathematical-physics are therefore, actually solved through integral equations, and in particular, the elliptic equations are often analysed through non-homogeneous Fredholm type integral equations of the second kind.

In general, the solution of an integral equation by purely analytical methods is not always possible, recourse, therefore, is taken to numerical methods.

In this chapter we have presented the numerical solution of two such integral equations. They are both Fredholm type non-homogeneous integral equations of the second kind, and we have taken them from the analytical works of Sneddon (1962 a) and Srivastav (1963). Their kernels are interesting and they have been briefly explained in the next section. Their numerical evaluation is, in fact, the special feature of this chapter. In solving the integral equation itself, we have employed the direct method which according to Scarborough (1958) was suggested by Goursat and later developed by Nystrom. It replaces the given integral equation by a finite set of simultaneous equations in the unknown function at discrete points.

5.2. A brief description of the Integral equations

As already pointed out, two integral equations have been considered in this chapter. The importance of the integral equation I lies in the fact that the integrand of its kernel has a pole at its lower limit of integration, and the

singularity arising from it requires suitable analytic treatment. It has been done in two ways and their numerical results compare very well.

The kernel of integral equation II has the same form of singularity as that of integral equation I, but in addition it is of special computational interest as well. The integrand of its kernel involves subtracting two functions which are close to each other in magnitude.

The loss of significant figures in this way is one of the serious sources of error in numerical work. The technique adopted here to treat such awkward situation is of general applicability and when a digital computer is available, it is convenient as well. It is interesting to see that in this way, accurate results have actually been obtained.

Both the integral equations are typical examples of their respective types. The integrations involved in the kernels have been evaluated using the standard Gauss-type quadrature formulae, while the integrations in the actual integral equation have performed by suitable Newton-Cotes-type formulae. The choice of these quadrature formulae has the basis that while Gauss-type formulae are well known for their high order accuracy, the Newton-Cotes type formulae are best suited when values of the wanted function are required at equal intervals of the variable.

5.3. Integral Equation I

Sneddon (1962 a) applies the dual series relations technique to the determination of the electrostatics field inside an infinite hollow- earthed cylinder of radius a containing an electrified circular disc of unit radius situated with its centre on the axis of the cylinder and its plane perpendicular to that axis. The problem is eventually reduced to that of solving a Fredholm integral equation of the second kind for the auxiliary function $g(t)$. It is

$$g(t) = 1 + \int_0^1 g(n) Kk(t, u) dn, 0 \leq t \leq 1 \quad (5.3.1)$$

where the kernel is

$$K(t, u) = \frac{4}{a\pi^2} \int_0^\infty \frac{K_0(w)}{I_0(w)} \cosh\left(\frac{tw}{a}\right) \cosh\left(\frac{uw}{a}\right) dw \quad (5.3.2)$$

where $K_0(w)$ and $I_0(w)$ are modified Bessel functions.

It is easy to see that the integrand in the kernel has a pole at the lower limit, because

$$\frac{K_0(x)}{I_0(x)} \rightarrow \infty, \text{ as } x \rightarrow 0$$

But we can subtract out the singularity using the fact

$$K_0(x) = -(\ln x - \ln 2 + r)I_0(x) + \sum_{r=1}^{\infty} \frac{(x/2)^{2r}}{LrLr} \left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{r}\right) \quad (5.3.3)$$

where γ is the Euler's constant (=0.5772156649.....)

we can write (5.3.3) as

$$\frac{K_0(x)}{I_0(x)} + \ln x = \ln 2 + r + \frac{1}{I_0(x)} \sum_{r=1}^{\infty} \frac{(x/2)^{2r}}{LrLr} \left(1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{r}\right) \quad (5.3.4.)$$

Remembering that
$$I_0(x) = \sum_{r=0}^{\infty} \frac{(x/2)^{2r}}{LrLr+1} \quad (5.3.5)$$

We can see that the right hand side of (5.3.4) is now well behaved at $x = 0$. On this basis we can write the kernel

$$\begin{aligned} K(t, n) &= \frac{4}{a\pi^2} \int_0^{\infty} \frac{K_0(w)}{I_0(w)} \cosh \frac{tw}{a} \cosh \frac{uw}{a} dw \quad (5.3.6) \\ &= \frac{4}{a\pi^2} (P - Q + R + S) \end{aligned}$$

Where

$$P = \int_0^{\infty} \frac{K_0(w) + (\ln w)I_0(w)}{I_0(w)} \cosh \frac{tw}{a} \cosh \frac{uw}{a} dw \quad (5.3.7)$$

$$Q = \int_0^{\infty} (\ln w) \cosh \frac{tw}{a} \cosh \frac{uw}{a} dw \quad (5.3.8)$$

$$R = \int_0^{\infty} \frac{K_0(w)}{I_0(w)} \cosh \frac{tw}{a} \cos \frac{uw}{a} dw \quad (5.3.9)$$

$$\text{and } S = \int_q^\infty \frac{K_0(w)}{I_0(w)} \cosh \frac{tw}{a} \cosh \frac{uw}{a} dw \quad (5.3.10)$$

We first attend to the integral Q. We can evaluate it by term-by-term integration of the power series expansion of the integrand, thus

$$\begin{aligned} Q &= \int_0^1 \sum_{m=0}^{\infty} \left[\frac{t^{2m}}{a^{2m} L_{2m}} \left(\sum_{n=0}^{\infty} \frac{w^{2m+2n} u^{2n}}{a^{2n} L_{2n}} \right) \right] \ln w dw \\ &= - \sum_{m=0}^{\infty} \frac{t^{2m}}{a^{2m} L_{2m}} \sum_{n=0}^{\infty} \frac{u^{2n}}{(2m+2n+1)^2 a^{2n} L_{2n}} \end{aligned} \quad (5.3.11)$$

Next, the integrations in P, R and S can be obtained by the Gauss-type quadrature formulae, using the zeros and weight given in AMS37 (1954). To suit limits of integration appropriate to these zeros and weights we make a change in the independent variable. Accordingly, we obtain

$$P = \frac{1}{2} \int_{-1}^{+1} \frac{K_0\left(\frac{z+1}{2}\right) + \left(\ln \frac{z+1}{2}\right) I_0\left(\frac{z+1}{2}\right)}{I_0\left(\frac{z+1}{2}\right)} \cosh \frac{(z+1)t}{2a} \cosh \frac{(z+1)u}{2a} dz \quad (5.3.12)$$

$$R = \frac{q-1}{2} \int_{-1}^{+1} \frac{K_0\left[\frac{a-1}{2}z + \frac{a+1}{2}\right]}{I_0\left[\frac{a-1}{2}z + \frac{a+1}{2}\right]} \cosh \frac{t}{a} \left(\frac{a-1}{2}z + \frac{a+1}{2}\right) \cosh \frac{u}{a} \left(\frac{a-1}{2}z + \frac{a+1}{2}\right) dz \quad (5.3.13)$$

$$S = \frac{a}{2a-t-u} \int_0^\infty e^{-z} \left[\frac{K_0\left(\frac{az}{2a-t-u} + q\right)}{I_0\left(\frac{az}{2a-t-u} + q\right)} \right] \cos \frac{t}{a} \left(\frac{az}{2a-t-u} + q\right) \cos \frac{u}{a} \left(\frac{az}{2a-t-u} + q\right) dz \quad (5.3.14)$$

In the determination q the method of trial and error is best suited. It was found that q = 8 is a right value ; and it has been used in evaluating R and S above.

5.31. The Numerical Procedure

In order to determine $g(t)$ in (5.3.1) at equidistant values of t , we cannot use the Gauss-Legendre quadrature formula to evaluate the integration involved. Instead we use a Newton-Cotes formula as pointed out earlier. We divide the range of integration, i.e. 0 to 1, into eight equal intervals so that we could use the formula

$$\int_0^1 p dx = \frac{2h}{45} [7p_0 + 32p_1 + 12p_2 + 32p_3 + 14p_4 + 32p_5 + 12p_6 + 32p_7 + 7p_8] \quad (5.3.15)$$

where P_r are the ordinates and $h = 1/8$ in this case.

This division of the range of integration leads us to the finding out of the Kernel $K(t, u)$ at points which are clearly the different combinations of the values of t and u as picked up from $t, u = \frac{r}{8} t, u$, where $r = 0, 1, 2, \dots, 8$.

The rest of the procedure is straight forward and the values of the integrals P, Q, R and S for $a = 1.2, 1.3, 1.4$ have been tabulated in appendices A(5)I, A(5)II, A(5)III, A(5)IV; the values of the kernel $K(t, u)$ have been tabulated in appendix A(5)V and finally the wanted function $g(t)$ has been tabulated in appendix A(5)VI.

5.32. The Alternate Approach

The kernel as presented in equation (5.3.2) has a comparatively simple form for straight forward evaluation as shown above. However, on applying a short analysis we can write the kernel in another form, viz.

$$K(t, u) = \frac{1}{a\pi^2} \left[k\left(2 + \frac{t+u}{a}\right) + k\left(2 - \frac{lt-ul}{a}\right) + k\left(2 + \frac{lt-ul}{a}\right) \right] \quad (5.3.16)$$

$$\text{where } k(\zeta) = \int_0^\infty \frac{e^{-\zeta z}}{I_0 z} \frac{e^{2z} K_0(z)}{dz}, \frac{2(a-1)}{a} < \zeta < \frac{2(a+1)}{a} \quad (5.3.17)$$

To evaluate $k(\zeta)$ we write

$$\begin{aligned} k(\zeta) &= \int_0^\infty e^{-\zeta z} \frac{e^{2z} K_0(z)}{I_0(z)} dz \\ &= A + B - C \end{aligned} \quad (5.3.18)$$

$$\text{where } A = \int_0^1 e^{(2-\zeta)z} \frac{K_0(z)(\ln z)}{I_0(z)} dz \quad (5.3.19)$$

$$B = \int_0^1 e^{(2-\zeta)z} \ln z dz \quad (5.3.20)$$

$$\text{and } C = \int_1^\infty e^{-\zeta z} \frac{e^{2z} K_0(z)}{I_0(z)} dz \quad (5.3.21)$$

The evaluation of the integrals A and C presents no difficulty. After a change of the independent variable they could be written as

$$A = \frac{1}{2} e^{\frac{1-\zeta}{2}} \int_{-1}^{+1} e^{t\left(\frac{1-\zeta}{2}\right)} \left[\frac{K_0\left(\frac{t+1}{2}\right) + I_0\left(\frac{t+1}{2}\right) \ln\left(\frac{t+1}{2}\right)}{I_0\left(\frac{t+1}{2}\right)} \right] dt \quad (5.3.22)$$

$$C = \frac{e^{2-\zeta}}{\zeta} \int_0^\infty e^{-t} \left[\frac{e^{2t/\zeta} K_0\left(\frac{t}{\zeta} + 1\right)}{I_0\left(\frac{t}{\zeta} + 1\right)} \right] dt \quad (5.3.23)$$

which can now be evaluated using the Gauss-Legendre and Gauss-Laguerre formulae of quadrature. It is in fact the evaluation of B, which presents some difficulty, for ordinarily, it diverges for $\zeta \neq 2$. But on using the formula in S_{324} , case 81a, page 81, Integraltafel Springer Verlag, 1958, we can write

$$B = \int_0^1 e^{(2-\zeta)z} \ln z dz = \frac{-1}{2-\zeta} \int_0^1 \frac{-1 + e^{(2-\zeta)z}}{z} dz, (\zeta \neq 2)$$

On changing the independent variable we can write

$$B = \frac{-1}{2-\zeta} \int_{-1}^{+1} \frac{-1 + e^{(1-\zeta/2)(1+t)}}{t+1} dt, (\zeta \neq 2)$$

We can now evaluate it for all $\zeta \neq 2$ using the Gaussian formula of quadrature.

However, for the particular case when $\zeta = 2$, we have from (5.3.20) direct

$$B = \int_0^1 \ln z dz = -1$$

With this analysis complete, we can evaluate A, B, C and therefore $k(\xi)$ for all ξ we require hitherto.

5.3.3 The Numerical Procedure

To use formula (5.3.15) for evaluating the integration in (5.3.1) we require the value of $k(S)$ for 180 values of S ; because according to (5.3.16) S has four forms and as each form admits of all possible combinations of t and u as picked up from $t, u = r/8, r = 0, 1, 2, \dots, 8$. We have forty-five combinations belonging to each form. We thus reduce equation (5.3.1) to a set of linear simultaneous algebraic equations which on solution yield the values of the wanted function $g(t)$ for $t = r/8, r = 0, 1, 2, \dots, 8$.

In the actual computation we took $a = 5/3$ this time, and we evaluate the kernel $K(t, u)$ both from its original form (5.3.2) and the modified form (5.3.17). The results were in agreement.

The Kernel has been recorded in appendix A(5)V and the wanted function $g(t)$ in appendix A(5)VI.

5.4 Integral Equation II

Srivastav (1963) conceives a long right circular cylinder of cross-sectional radius a , one end of which is linked rigidly to a rigid foundation. A circular disc of radius unity is imagined in the plane face on the other end, such that the centre of the disc lies on the axis of the cylinder. If the disc is made to rotate through a certain given angle, torsional oscillations are produced in the cylinder. While investigating into these oscillations, through the application of dual relations series, he finally seeks the auxiliary function $g(t)$ satisfying the Fredholm integral equation of the second kind;

$$g(t) = t + \int_0^{\infty} g(u)k(t, u)du \quad (5.4.1)$$

the kernel of the equation being

$$K(t, u) = \frac{16}{\pi} ut(1 - 2\delta) + \frac{4}{\pi^2} \int_0^{\infty} \frac{K_2(y)}{I_2(y)} [8utI_2(y) - \sinh(ty)\sinh(uy)] dy \quad (5.4.2)$$

where $c = 1/a$, $8 =$ length of cylinder and $\mathbf{K}_2 y$ and $\mathbf{I}_2 y$ are modified Bessel functions.

It is easy to see that the integrand in the kernel (5.4.2) has the same form of singularity as was encountered in integral equation I above. Hence, we can treat it in exactly the same way, and there is nothing exciting about it so far. But, as pointed out earlier, the kernel is really interesting from a computational point of view. To appreciate it, Let us have a look at the function within square brackets in the integrand in (5.4.2). Using Taylor expansions and setting

$$P = t + u$$

$$Q = t - u$$

$$8ut\mathbf{I}_2(y) = 2(p^2 - q^2)\mathbf{I}_2(y)$$

$$= 2(p^2 - q^2) \sum_{s=0}^{\infty} \frac{(y/2)^{2+2s}}{\angle s \angle s + 2}$$

$$= \frac{1}{4}(p^2 - q^2)y^2 + \frac{1}{48}(p^2 - q^2)y^4 + \frac{1}{1536}(p^2 - q^2)y^6 + \dots$$

and

$$\sinh(ty) \sinh(uy) = \frac{1}{2}(\cosh py - \cosh qy)$$

$$= \frac{1}{4} \left(p^2 - q^2 y^2 + \frac{1}{48}(p^4 - q^4)y^4 + \frac{1}{1440}(p^6 - q^6)y^6 + \dots \right)$$

Now, it is obvious that the principal terms in the two expansions are the same, consequently, when one is subtracted from the other, a wholesale loss of significant figures is bound to happen. Unless it is checked, the results would be terribly inaccurate. The best way to deal with this sort of situation is to cancel these terms analytically, and to use the rest of the power series in place of the function in question. Thus we replace the function

$$8utI_2(y) - \sinh(ty) \sinh(uz)$$

By its power series equivalent

$$\frac{ut}{2} \sum_{s=0}^{\infty} \frac{2^{-2s} \cdot y^{2s+4}}{\angle s + 1 \angle s + 3} - u \sum_{s=0}^{\infty} \frac{t^{2s+3} \cdot y^{2s+4}}{\angle 2s + 3} - \frac{u^3}{\angle 3} \sum_{s=0}^{\infty} \frac{t^{2s+1} \cdot y^{2s+4}}{\angle 2s + 1} - \frac{u^5}{\angle 5} \sum_{s=0}^{\infty} \frac{t^{2s+1} \cdot y^{2s+6}}{\angle 2s + 1} - \dots \quad (5.4.3)$$

The technique works efficiently with digital computers.

Thus having defined the basis of our approach to carry out the integration involved in the kernel (5.4.2) in the following manner,

$$K(t, u) = \frac{16}{\pi} ut(1 - 2\delta) + \frac{4}{\pi^2} [A - B + C - D - E] \quad (5.4.4)$$

where

$$\begin{aligned} A &= \int_0^1 \frac{K_2(y) + (\ln y)I_2(y)}{I_2(y)} [8utI_2(y) - \sinh(ty) \sinh(uy)] dy \\ &= \frac{1}{2} \int_{-1}^{+1} \frac{K_2\left(\frac{p+1}{2}\right) + \left(\ln \frac{p+1}{2}\right)}{I_2\left(\frac{p+1}{2}\right)} \left[8utI_2\left(\frac{p+1}{2}\right) - \sinh \frac{t(p+1)}{2} \sinh \frac{u(p+1)}{2} \right] dp \quad (5.4.5) \end{aligned}$$

$$B = \int_0^1 [8utI_2(y) - \sinh(ty) \sinh(uy)] \ln y dy$$

$$\begin{aligned} &= \int_0^1 \left[\frac{ut}{2} \sum_{s=0}^{\infty} \frac{2^{-2s} \cdot y^{2s+4}}{\angle s + 1 \angle s + 3} - u \sum_{s=0}^{\infty} \frac{t^{2s+3} \cdot y^{2s+4}}{\angle 2s + 3} - \frac{u^3}{\angle 3} \sum_{s=0}^{\infty} \frac{t^{2s+1} \cdot y^{2s+4}}{\angle 2s + 1} - \dots \right] \ln y dy \\ &= ut \left[-\frac{1}{2} \sum_{s=0}^{\infty} \frac{1}{(2s+5)^2 \cdot 2^{2s} \angle s + 1 \angle s + 3} + \sum_{s=0}^{\infty} \frac{t^{2s+2}}{(2s+5)^2 \angle 2s + 3} + \frac{u^2}{\angle 3} \sum_{s=0}^{\infty} \frac{t^{2s}}{(2s+5)^2 \angle 2s + 1} + \dots \right] \quad (5.4.6) \end{aligned}$$

by term by term integration.

$$C = 8ut \int_1^{\infty} K_2(y) dy$$

$$D = \int_1^r \frac{K_2(y)}{I_2(y)} \sinh(ty) \sinh(uy) dy$$

and

$$E = \int_r^{\infty} \frac{K_2(y)}{I_2(y)} \sinh(ty) \sinh(uy) dy$$

To evaluate C, we use the fact

$$K_0(y) + K_2(y) = -2K_1'(y).$$

integrating it , we obtain

$$\int_1^{\infty} K_0(y) dy + \int_1^{\infty} K_2(y) dy = 2K_1(1)$$

or

$$\int_1^{\infty} K_2(y) dy = 2K_1(1) - \int_1^{\infty} K_0(y) dy$$

$$= 2(0.6019072302) -$$

$$(0.328286478)^{1*}$$

$$= 0.8875527982$$

Thus

$$C = 8ut (0.875527982) \quad (5.4.7)$$

Again, by the method of trial and error it was found that $r=4$ is a suitable value. Hence we write

$$D = \int_1^4 \frac{K_2(y)}{I_2(y)} \sinh(ty) \sinh(uy) dy$$

$$= \frac{3}{2} \int_{-1}^{+1} \frac{K_2\left(\frac{3x+5}{2}\right)}{I_2\left(\frac{3x+5}{2}\right)} \sinh \frac{t(3x+5)}{2} \sinh \frac{u(3x+5)}{2} dx$$

(5.4.8)

and lastly

¹ $\int_1^{\infty} K_0(y) dy = 0.328286478$; (Phil. Mag. Vol 20, 1935(7) Bickley and Nayler, pp. 343-347)

$$\begin{aligned}
E &= \int_4^\infty \frac{K_2(y)}{I_2(y)} \sinh(ty) \sinh(uy) dy \\
&= \int_0^\infty \frac{K_2(x+4)}{I_2(x+4)} \sinh t(x+4) \sinh u(x+4) dx
\end{aligned}$$

When x is large, the integrand $e^{-2(x+4)} e^{(x+4)(u+t)}$

$$e^{-4(2-u-t)} e^{-(2-u-t)x}$$

To arrest its exponential like behavior, we multiply $e^{(2-u-t)x}$, hence

$$E = \int_0^\infty e^{-(2-u-t)x} \left[e^{(2-u-t)x} \frac{K_2(x+4)}{I_2(x+4)} \sinh t(x+4) \sinh u(x+4) \right] dx \quad (5.4.9)$$

$$= \frac{1}{2-u-t} \int_0^\infty \left[e^z \frac{K_2\left(4 + \frac{z}{2-t-u}\right)}{I_2\left(4 + \frac{z}{2-u-t}\right)} \sinh t\left(4 + \frac{z}{2-t-u}\right) \sinh u\left(4 + \frac{z}{2-t-u}\right) \right] dz$$

On replacing the function within square brackets in (5.4.5) we can evaluate A using the Gauss-Legendre quadrature. Similarly D and E can also be evaluated. Thus we can obtain the Kernel from (5.4.4)

5.41. The Numerical Procedure

In actual computation we took $\delta = 10$ units, and as many as four cases were considered, which are $C = 0.4, 0.5, 0.6, 0.7$. According to the different values of C, different Newton-Cotes formulae were taken for performing the integration in (5.4.1). These formulae are listed in appendix A(5) VII. The values of the integrals A, B, C, D, E and the kernel, are tabulated in appendix A(5) VIII. These values can cater for the cases $c = 0.4, 0.5, 0.6, 0.7$. Finally, the wanted function $g(t)$ is recorded in appendix A(5) IX.

5.5 Check on the Accuracy of Results

It is important that numerical results should be checked by a suitable method . Accordingly, all the results obtained above, in both the integral equations, have been checked and found to be reliable.

In both the integral equations it is the evaluation of the kernels which is pretty involved . Hence we need be sure about their accurate evaluation. We have already pointed out that Gaussian quadrature has been used in determining the kernels. This sort of quadrature is highly accurate and the best way to check its results is to re-evaluate the integral using a different number of ordinates this time. If the two results agree then only they are reliable This well-known method was employed in both the integral equations and checks were made for certain values of t and u, and the trial and error method referred to above for the determination of the limits q and r is infact this basis.

In addition to this , the kernel in integral equation I has been evaluated in an alternate way as well; and the agreement in the two results indicates their reliability.

On differencing the wanted function g(t) of integral equation I, we find it to be well-behaved, which is in accordance with the estimates of Sneddon.

To check the wanted function g(t) of integral equation II, we find that Srivastav points out that it is possible to approximate the function g(t) by a polynomial in odd powers of t. Hence we write

$$g(t) = \gamma_1 \left(\frac{t}{c} \right) + \gamma_2 \left(\frac{t}{c} \right)^3 + \gamma_3 \left(\frac{t}{c} \right)^5 \quad (5.5.1)$$

and

$$g(t) = \Gamma_1 \left(\frac{t}{c} \right) \quad (5.5.2)$$

as well, as good fits. If a least square fitting be done, we find that the coefficients $\gamma_1, \gamma_2, \gamma_3$ and Γ_1 are as follows

No. = 0.6	Coeffs	c = 0.4	c = 0.5	c
1	Γ_1	+0.13103	+0.09939	+0.07497
2	γ_1	+0.13171	+0.08929	+0.07806

3	γ_2	-0.00091	+0.03374	-0.00419
4	γ_3	-0.00002	-0.02332	-0.00031

Further , on multiplying (5.5.1) and (5.5.2) by t and integrating from 0 to c we obtain

$$\int_0^c tg(t)dt = c^2 \left(\frac{\gamma_1}{3} + \frac{\gamma_3}{5} + \frac{\gamma_5}{7} \right) \quad (5.5.3)$$

and

$$\int_0^c tg(t)dt = \frac{c^2 \Gamma_1}{3} \quad (5.5.4)$$

respectively.

We are now in a position to test the accuracy of our results. We can obtain the values of the left hand sides i.e. $\int_0^c tg(t)dt$ from appendix A(5) X, whereas the right hand sides can be computed with the help of the table 1. Having done this we find that the left and right hand sides are in agreement. We record their values in table 2.

TABLE 2

FUNCTION	c =0.4	c = 0.5	c = 0.6
RHS or LHS	0.00699	0.00829	0.00905

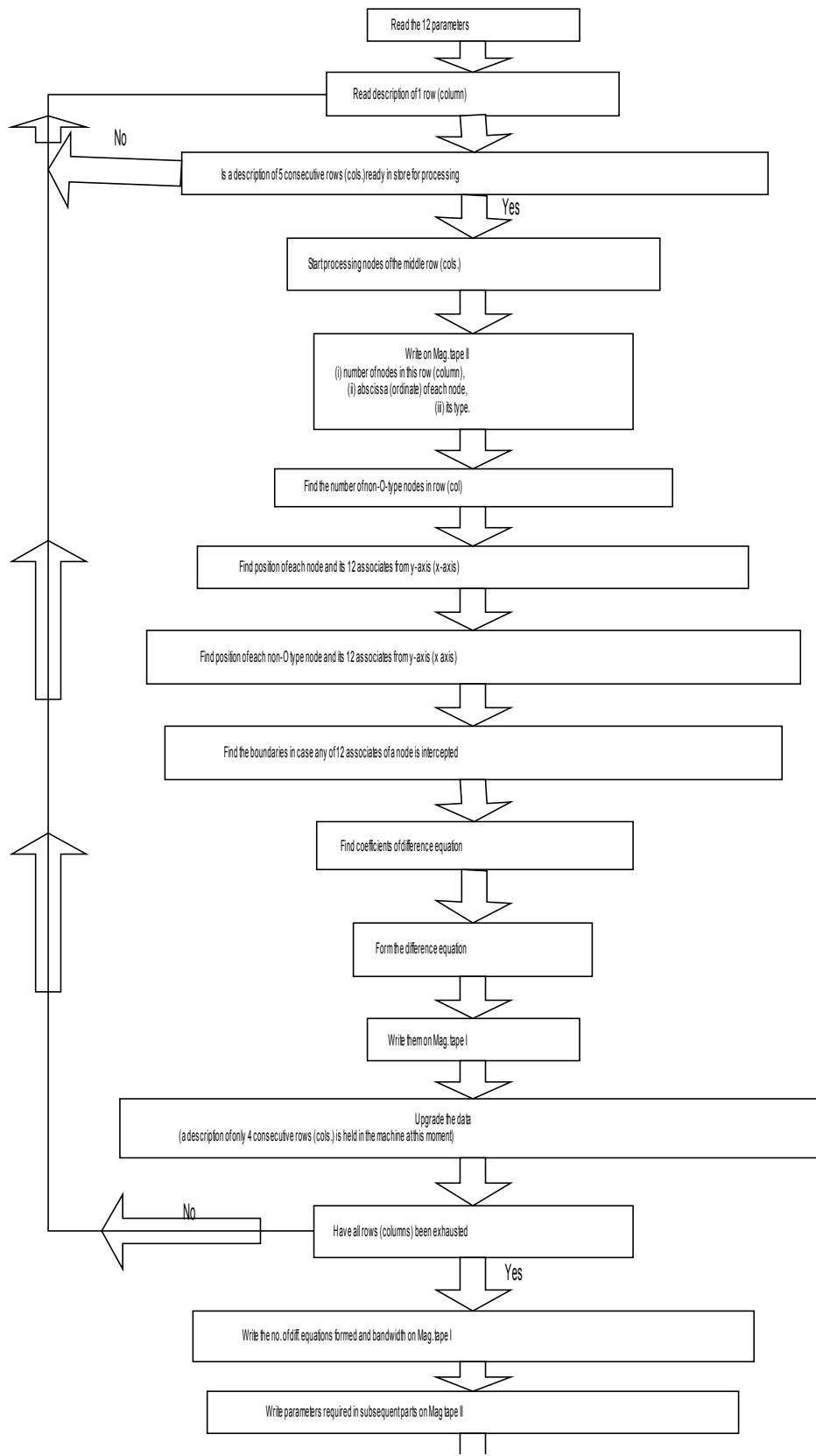
This indicates that our numerical results are in accordance with the analytical estimates of Srivastav.

Concluding Remarks

The numerical work was done on DEUCE using single length precision; for evaluating Bessel functions; standard subroutines of the DEUCE library were used.

APPENDIX A(2) I

Of the four parts into which the programme is divided, part I is the most important. Hence a flow chart of only this part is given on the following page. In accordance with the relaxation patterns developed in chapter I, a typical node has 12 points associated with it . In the flow chart they have been referred to as the 12 associates of the node. In addition the mag. Types DGO10002 and DGO10003 have been referred to as Mag. Tapes I and II respectively. The chart indicates the major step of the programme.



APPENDIX A(2) II

A failure could happen either through a fault in the procedures or in the data. Whenever it happens, message is sent out by the monitor typewriter. It gives the node, the row (column) to which it belongs, as also the failure number. Description of the failures is ;

Failures 1-4 refer to a node whose one arm is interrupted by a Neumann boundary;

1. Users procedure DERV is incorrectly evaluating the angle which the normal gradient makes with the x-axis.
2. Incorrect description of the mesh around this node
3. $m < 0$, the extra node is not available in the region
4. $m > 0$, the extra node is not available in the region.

Failures 5-9 refer to a node whose two arms are intercepted by the Neumann boundary;

5. None of the two extra nodes is available in the region.
6. Users procedure DERV is incorrectly calculating the angle.
- 7,8,9. Incorrect description of mesh around this node.

Failure 10 . Incorrect description of mesh around a node whose arm(s) is (are) interrupted by Dirichlet boundary.

APPENDIX A(3) I

m-JJ

Case 1.

$B_0 = -1415.42$

0 0 0 0 0 0 0 0 0

0 0 0

0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0
2	2	2	1	1	1	1	0	0	0
2	2	2	2	2	1	1	1	0	0
3	3	3	2	2	2	1	1	0	0
4	4	4	3	3	2	2	1	0	0
5	5	4	4	3	3	2	1	1	0
6	6	6	5	4	4	3	2	1	0
8	8	7	6	6	4	3	2	1	0
10	10	9	8	7	6	4	3	1	0
13	13	12	11	9	7	6	4	2	0
17	16	15	14	12	9	7	5	2	0
22	21	19	17	15	12	9	6	3	0
28	26	24	22	19	15	11	8	4	0
35	34	31	28	24	19	14	10	5	0
45	43	40	35	30	25	18	12	6	0
57	54	50	45	39	31	24	15	8	0
72	69	64	57	49	40	30	20	10	0
91	87	81	73	62	51	38	25	12	0
116	111	103	92	79	64	48	32	15	0
147	141	131	117	101	82	61	40	20	0
186	178	166	149	128	104	78	51	25	0
235	226	211	189	163	132	99	65	32	0
297	286	267	241	208	169	126	83	40	0
373	360	339	307	265	216	161	105	51	0
467	453	429	392	340	276	204	132	63	0
578	565	542	504	439	355	259	165	78	0
707	697	680	656	572	460	325	201	94	0
850	843	831	808	755	608	399	235	107	0

1000	1000	1000	1000	1000	1000	444	252	113	0
------	------	------	------	------	------	-----	-----	-----	---

APPENDIX A(3) II

m-JJ

Case 2.

$B_0 = -1465.42$

$B_1 = 42.60$

0	0	0	0	0	0	0	0	0	0
1	1	1	0	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0
2	2	1	1	1	1	1	0	0	0
2	2	2	2	2	1	1	1	0	0
3	3	3	2	2	2	1	1	0	0
4	4	3	3	3	2	2	1	0	0
5	5	4	4	3	3	2	1	1	0
6	6	6	5	4	4	3	2	1	0
8	8	7	6	6	4	3	2	1	0
10	10	9	8	7	6	4	3	1	0
13	13	12	10	9	7	5	4	2	0
17	16	15	13	11	9	7	4	2	0
21	20	19	17	14	12	9	6	3	0
27	26	24	22	18	15	11	7	4	0
34	33	31	27	24	19	14	9	4	0
44	42	39	35	30	24	18	12	6	0
56	53	50	44	38	31	23	15	7	0
71	68	63	56	48	39	29	19	9	0
90	86	80	71	61	50	37	24	12	0
114	109	101	91	78	63	47	31	15	0
115	139	129	115	99	80	60	40	19	0
183	176	164	147	126	102	77	50	24	0
232	223	208	186	160	130	97	64	31	0
293	282	263	237	204	166	124	81	39	0
369	356	334	302	261	211	157	102	49	0
462	448	424	387	334	270	199	128	62	0
574	561	536	497	431	347	251	160	76	0

704	693	674	648	562	449	314	194	90	0
847	841	828	802	742	592	379	225	103	0
1000	1000	1000	1000	1000	1000	427	241	108	0

APPENDIX A(3) III

m-JJ

Case 3.

$B_0 = -1421.27$

0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0
2	2	2	1	1	1	1	0	0	0
2	2	2	2	2	1	1	1	0	0
3	3	3	2	2	2	1	1	0	0
4	4	3	3	3	2	2	1	0	0
5	5	4	4	3	3	2	1	1	0
6	6	6	5	4	4	3	2	1	0
8	8	7	6	6	4	3	2	1	0
11	10	9	8	7	6	4	3	1	0
13	13	12	11	9	7	6	4	2	0
17	16	15	14	12	9	7	5	2	0
22	21	19	17	15	12	9	6	3	0
28	26	24	22	19	13	11	8	4	0
35	34	31	28	24	19	14	10	5	0
44	43	40	35	30	25	16	12	6	0
56	54	50	45	38	31	23	15	8	0
72	69	64	57	49	40	30	20	10	0
91	87	81	72	62	50	38	25	12	0
116	111	103	92	79	64	48	32	15	0
147	140	130	117	100	82	61	40	20	0
186	178	166	149	128	104	78	51	25	0
235	226	210	189	163	132	99	65	32	0
296	285	267	241	207	168	126	83	40	0
373	360	338	307	265	215	160	105	50	0
466	453	428	392	340	275	204	132	63	0

578	565	542	503	438	354	258	144	78	0
707	697	679	655	572	459	324	200	93	0
849	843	831	808	754	607	396	234	107	0
1000	1000	1000	1000	1000	1000	441	250	112	0

APPENDIX A(3) IV

m-JJ

Case 4

$B_0 = -140085$

$B_0 = 58.04$

0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0
2	2	2	1	1	1	1	0	0	0
2	2	2	2	2	1	1	1	0	0
3	3	3	2	2	2	1	1	0	0
4	4	4	3	3	2	2	1	0	0
5	5	4	4	3	3	2	1	1	0
6	6	6	5	4	4	3	2	1	0
8	8	7	6	6	4	3	2	1	0
11	10	9	8	7	6	4	3	1	0
13	13	12	11	10	7	6	4	2	0
17	16	15	14	12	10	7	5	2	0
22	21	19	17	15	12	9	6	3	0
28	26	24	22	19	15	11	8	4	0
35	34	31	28	24	19	14	10	5	0
45	43	40	35	30	25	18	12	6	0
57	54	50	45	39	31	24	15	8	0
72	69	64	57	49	40	30	20	10	0
91	88	81	73	62	51	38	25	12	0
116	111	103	92	79	64	48	32	15	0
147	141	131	117	101	82	62	40	20	0
186	178	166	149	128	104	78	51	25	0
235	226	211	190	163	133	100	65	32	0

297	286	267	241	208	169	127	83	40	0
373	361	339	307	266	216	161	105	51	0
467	453	429	392	341	277	205	133	64	0
578	565	542	504	439	356	261	166	79	0
707	699	679	655	572	461	329	204	95	0
849	843	831	807	751	610	407	239	109	0
1000	1000	1000	1000	1000	1000	450	256	115	0

APPENDIX A(3)V

m-JJ

Case 5

$B_0 = -1418.74$

0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0
2	2	2	1	1	1	1	0	0	0
2	2	2	2	2	1	1	1	0	0
3	3	3	2	2	2	1	1	0	0
4	4	3	3	3	2	2	1	0	0
5	5	4	4	3	3	2	1	1	0
6	6	6	5	4	4	3	2	1	0
8	8	7	6	6	4	3	2	1	0
10	10	9	8	7	6	4	3	1	0
13	13	12	11	9	7	6	4	2	0
17	16	15	14	12	9	7	5	2	0
22	21	19	17	15	12	9	6	3	0
28	26	24	22	19	15	11	8	4	0
35	34	31	28	24	19	14	10	5	0
44	43	40	35	30	25	18	12	6	0
57	54	50	45	38	31	23	15	8	0
72	69	64	57	49	40	30	20	10	0
91	87	81	72	62	50	38	25	12	0
116	111	103	92	79	64	48	32	15	0
147	141	131	117	101	82	61	40	20	0
186	178	166	149	128	104	78	51	25	0

235	226	210	189	163	132	99	65	32	0
297	286	267	241	208	169	126	83	40	0
373	360	338	307	265	215	161	105	50	0
466	453	429	392	340	276	204	132	63	0
578	565	542	504	439	354	258	164	78	0
707	697	680	655	572	459	324	200	94	0
849	843	831	808	754	608	397	234	107	0
1000	1000	1000	1000	1000	1000	442	251	113	0

APPENDIX A(3) VI

m-JJ

Case 6.

$B_0 = -1432.60$

$B_0 = 32.93$

0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0
2	2	2	1	1	1	1	0	0	0
2	2	2	2	2	1	1	1	0	0
3	3	3	2	2	2	1	1	0	0
4	4	3	3	3	2	2	1	0	0
5	5	4	4	3	3	2	1	1	0
6	6	6	5	4	4	3	2	1	0
8	8	7	6	6	4	3	2	1	0
10	10	9	8	7	6	4	3	1	0
13	13	12	11	9	7	6	4	2	0
17	16	15	14	12	9	7	5	2	0
22	21	19	17	15	12	9	6	3	0
27	26	24	22	19	15	11	7	4	0
35	33	31	28	24	19	14	10	5	0
44	42	39	35	30	24	18	12	6	0
56	54	50	45	38	31	23	15	7	0
72	68	63	57	49	40	30	20	10	0
91	87	81	72	62	50	38	25	12	0

115	110	102	92	79	64	48	32	15	0
146	140	130	116	100	81	61	40	20	0
185	177	165	148	127	103	78	51	25	0
234	225	209	188	162	132	99	65	31	0
295	284	266	240	206	168	125	82	40	0
372	359	337	305	264	214	160	104	50	0
465	451	427	390	338	274	202	131	63	0
576	563	540	501	436	352	256	163	77	0
706	695	677	652	568	456	321	199	93	0
848	842	830	805	748	602	393	232	106	0
1000	1000	1000	1000	1000	1000	439	248	112	0

APPENDIX A(3) VII

m-JJ

Case 7.

$B_0 = -1431.74$

0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0
2	2	2	1	1	1	1	0	0	0
2	2	2	2	2	1	1	1	0	0
3	3	3	2	2	2	1	1	0	0
4	4	3	3	3	2	2	1	0	0
5	5	4	4	3	3	2	1	1	0
6	6	6	5	4	4	3	2	1	0
8	8	7	6	6	4	3	2	1	0
10	10	9	8	7	6	4	3	1	0
13	13	12	11	9	7	6	4	2	0
17	16	15	14	12	9	7	5	2	0
22	21	19	17	15	12	9	6	3	0
28	26	24	22	19	15	11	8	4	0
35	33	31	28	24	19	14	10	5	0
44	42	39	35	30	24	18	12	6	0
56	54	50	45	38	31	23	15	8	0

72	68	64	57	49	40	30	20	10	0
91	87	81	72	62	50	38	25	12	0
115	110	102	92	79	64	48	32	15	0
146	140	130	117	100	81	61	40	20	0
185	178	165	148	127	104	78	51	25	0
234	225	210	189	162	132	99	65	32	0
296	285	266	240	207	168	126	83	40	0
372	359	338	306	264	214	160	105	50	0
466	452	428	391	339	274	203	132	63	0
577	564	540	502	437	352	256	164	77	0
706	696	678	654	570	457	321	200	93	0
849	843	830	807	752	604	392	234	106	0
1000	1000	1000	1000	1000	1000	437	251	111	0

APPENDIX A(3) VIII

m-JJ

Case 8

$B_0 = -1440.16$

$B_1 = 36.99$

0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0
2	2	2	1	1	1	1	0	0	0
2	2	2	2	2	1	1	1	0	0
3	3	3	2	2	2	1	1	0	0
4	4	3	3	3	2	2	1	0	0
5	5	4	4	3	3	2	1	1	0
6	6	6	5	4	4	3	2	1	0
8	8	7	6	6	4	3	2	1	0
10	10	9	8	7	6	4	3	1	0
13	13	12	11	9	7	6	4	2	0
17	16	15	14	12	9	7	5	2	0
22	21	19	17	15	12	9	6	3	0
27	26	24	22	19	15	11	7	4	0

35	33	31	28	24	19	14	9	5	0
44	42	39	35	30	24	18	12	6	0
56	54	50	45	38	31	23	15	7	0
71	68	63	57	49	39	30	19	9	0
90	87	80	72	62	50	38	25	12	0
115	110	102	92	78	64	48	31	15	0
146	140	130	116	100	81	61	40	19	0
184	177	165	148	127	103	77	51	25	0
234	224	209	188	162	131	98	64	31	0
295	284	265	239	206	167	125	82	40	0
371	358	336	304	263	213	159	104	50	0
464	450	426	389	337	273	202	130	62	0
576	563	539	500	435	350	255	162	77	0
705	695	677	651	567	454	319	197	92	0
848	842	829	804	747	600	390	230	105	0
1000	1000	1000	1000	1000	1000	436	247	111	0

APPENDIX A(3) IX

m-Motz

$b_0 = -1433.66$

$b_0 = 117.55$

0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0
2	2	2	1	1	1	1	0	0	0
2	2	2	2	2	1	1	1	0	0
3	3	3	2	2	2	1	1	0	0
4	4	3	3	3	2	2	1	0	0
5	5	4	4	3	3	2	1	1	0
6	6	6	5	4	4	3	2	1	0
8	8	7	6	6	4	3	2	1	0
10	10	9	8	7	6	4	3	1	0
13	12	12	10	9	7	5	4	2	0
16	16	15	13	11	9	7	4	2	0
21	20	18	17	14	12	9	6	3	0

27	25	24	21	18	15	11	7	4	0
34	32	30	27	23	19	14	9	4	0
43	41	38	34	29	24	18	12	6	0
54	52	48	43	37	30	23	15	7	0
69	66	61	55	47	38	29	19	9	0
88	84	78	70	60	49	36	24	12	0
112	107	99	89	76	62	46	30	15	0
141	135	126	113	97	79	59	39	19	0
179	172	160	143	123	100	75	49	24	0
227	218	203	182	157	128	96	63	30	0
286	275	257	232	200	163	122	80	39	0
360	348	326	295	255	208	155	101	49	0
452	437	412	376	327	266	198	128	62	0
562	548	521	480	421	342	251	160	76	0
693	680	655	614	548	446	317	196	92	0
841	833	817	787	728	597	391	231	105	0
1000	1000	1000	1000	1000	1000	444	249	111	0

APPENDIX A(3) X

(Ignoring presence of singularity)

0	0	0	0	0	0	0	0	0	0
1	1	1	1	0	0	0	0	0	0
1	1	1	1	1	1	0	0	0	0
2	2	2	1	1	1	1	0	0	0
2	2	2	2	2	1	1	1	0	0
3	3	3	2	2	2	1	1	0	0
4	4	4	3	3	2	2	1	0	0
5	5	4	4	3	3	2	1	1	0
6	6	6	5	4	4	3	2	1	0
8	8	7	7	6	5	3	2	1	0
11	10	9	8	7	6	4	3	1	0
14	13	12	11	9	8	6	4	2	0
17	16	15	14	12	10	7	5	2	0
22	21	20	17	15	12	9	6	3	0

28	27	25	22	19	15	12	8	4	0
36	34	32	28	24	20	15	10	5	0
45	43	40	36	31	25	19	12	6	0
57	55	51	46	39	32	24	16	8	0
72	70	64	58	50	40	30	20	10	0
92	88	82	73	63	51	38	25	12	0
117	112	104	93	80	65	49	32	16	0
148	142	132	118	102	83	62	41	20	0
187	179	167	150	130	106	80	52	26	0
236	227	212	191	165	135	102	67	33	0
297	286	268	243	210	172	130	86	42	0
372	360	339	308	269	225	166	109	53	0
464	451	427	392	348	284	214	139	67	0
574	561	536	499	444	369	274	177	84	0
703	691	670	634	577	486	352	220	103	0
847	840	826	802	758	666	447	265	121	0
1000	1000	1000	1000	1000	1000	534	292	129	0

APPENDIX A(5) I

No.	t	U	VALUES OF INTEGRAL P		
			a = 1.2 a = 1.4	a = 1.3	
1	0.000	0.000	0.192291	0.192291	0.192291
2	0.000	0.125	0.192746	0.192678	0.192625
3	0.000	0.250	0.194112	0.193842	0.193628
4	0.000	0.375	0.194601	0.195790	0.195306
5	0.000	0.500	0.199627	0.198533	0.197668
6	0.000	0.625	0.203815	0.202089	0.200725
7	0.000	0.750	0.208995	0.206479	0.208995
8	0.000	0.875	0.215203	0.211730	0.20995
9	0.000	1.000	0.222484	0.217874	0.214252
10	0.125	0.125	0.193202	0.193067	0.192960
11	0.125	0.250	0.194573	0.194234	0.193966
12	0.125	0.375	0.196870	0.196188	0.195648
13	0.125	0.500	0.200108	0.198940	0.198015
14	0.125	0.625	0.204311	0.202506	0.201081
15	0.125	0.750	0.209509	0.206909	0.204860
16	0.125	0.875	0.215740	0.212176	0.209373
17	0.125	1.000	0.223047	0.218339	0.214643
18	0.250	0.250	0.195959	0.195412	0.194979
19	0.250	0.375	0.198281	0.197384	0.196675
20	0.250	0.500	0.201554	0.200161	0.199061
21	0.250	0.625	0.205802	0.203756	0.202151
22	0.250	0.750	0.211056	0.208204	0.205960
23	0.250	0.875	0.217353	0.213519	0.210508
24	0.250	1.000	0.224740	0.219738	0.215820
25	0.375	0.375	0.200643	0.199385	0.198393
26	0.375	0.500	0.203974	0.202204	0.200810
27	0.375	0.625	0.208298	0.205858	0.203940
28	0.375	0.750	0.213646	0.210369	0.207799
29	0.375	0.875	0.220056	0.215765	0.212407
30	0.375	1.000	0.227575	0.222080	0.217789
31	0.500	0.500	0.207388	0.205082	0.203271
32	0.500	0.625	0.211818	0.208813	0.206458
33	0.500	0.750	0.217298	0.213420	0.210387
34	0.500	0.875	0.223867	0.218390	0.215080
35	0.500	1.000	0.231573	0.225378	0.220560
36	0.625	0.625	0.216388	0.212644	0.209719
37	0.625	0.750	0.222040	0.217374	0.213739
38	0.625	0.875	0.228815	0.223033	0.218541
39	0.625	1.000	0.236764	0.229654	0.224149
40	0.750	0.750	0.227905	0.222257	0.217872
41	0.750	0.875	0.234936	0.228098	0.222808
42	0.750	1.000	0.243185	0.234935	0.228573
43	0.875	0.875	0.242274	0.234159	0.227905
44	0.875	1.000	0.250884	0.241252	0.233858
45	1.000	1.000	0.259919	0.248647	0.240032

APPENDIX A(5) II

No	t	U	VALUES OF INTEGRAL Q		
			a = 1.2	a = 1.3	a = 1.4
1	0.000	0.000	-1.000000	-1.000000	-1.000000
2	0.000	0.125	-1.000603	-1.000514	-1.000443
3	0.000	0.250	-1.002414	-1.002057	-1.001773
4	0.000	0.375	-1.005441	-1.004634	-1.003994
5	0.000	0.500	-1.009695	-1.008255	-1.007113
6	0.000	0.625	-1.015194	-1.012930	-1.011138
7	0.000	0.750	-1.021957	-1.018677	-1.016082
8	0.000	0.875	-1.030013	-1.025513	-1.021957
9	0.000	1.000	-1.039393	-1.033462	-1.028782
10	0.125	0.125	-1.001207	-1.001028	-1.000887
11	0.125	0.250	-1.003022	-1.002574	-1.002219
12	0.125	0.375	-1.006055	-1.005156	-1.004443
13	0.125	0.500	-1.010317	-1.008782	-1.007566
14	0.125	0.625	-1.015826	-1.013466	-1.011598
15	0.125	0.750	-1.022604	-1.019222	-1.016548
16	0.125	0.875	-1.030675	-1.026070	-1.022432
17	0.125	1.000	-1.040074	-1.034032	-1.029267
18	0.250	0.250	-1.004848	-1.004127	-1.003557
19	0.250	0.375	-1.007898	-1.006722	-1.005791
20	0.250	0.500	-1.012186	-1.010367	-1.008928
21	0.250	0.625	-1.017727	-1.015074	-1.012976
22	0.250	0.750	-1.024544	-1.020859	-1.017948
23	0.250	0.875	-1.032664	-1.027741	-1.023857
24	0.250	1.000	-1.042119	-1.035744	-1.030722
25	0.375	0.375	-1.010979	-1.009338	-1.008041
26	0.375	0.500	-1.015308	-1.013013	-1.011200
27	0.375	0.625	-1.02094	-1.017760	-1.015278
28	0.375	0.750	-1.027788	-1.023593	-1.020285
29	0.375	0.875	-1.035988	-1.030533	-1.026238
30	0.375	1.000	-1.045536	-1.038604	-1.033152
31	0.500	0.500	-1.019697	-1.016731	-1.014391
32	0.500	0.625	-1.025369	-1.021533	-1.018510
33	0.500	0.750	-1.032347	-1.027434	-1.023568
34	0.500	0.875	-1.040660	-1.034456	-1.029580
35	0.500	1.000	-1.050340	-1.042621	-1.022681
36	0.625	0.625	-1.031140	-1.026406	-1.022681
37	0.625	0.750	-1.038241	-1.032395	-1.027804
38	0.625	0.875	-1.046699	-1.039522	-1.033894
39	0.625	1.000	-1.056550	-1.047809	-1.040969
40	0.750	0.750	-1.045492	-1.038493	-1.033008
41	0.750	0.875	-1.054131	-1.045749	-1.039193
42	0.750	1.000	-1.064192	-1.054188	-1.046379
43	0.875	0.875	-1.062985	-1.053159	-1.045492
44	0.875	1.000	-1.073297	-1.061778	-1.052811
45	1.000	1.000	-1.083904	-1.070607	-1.060283

APPENDIX A(5) III

No	T	U	VALUES OF INTEGRAL Q		
			a = 1.2 a = 1.4	a = 1.3	
1	0.000	0.000	-0.037511	0.175385	0.175385
2	0.000	0.125	0.177848	0.177482	0.177192
3	0.000	0.250	0.185352	0.183858	0.182677
4	0.000	0.375	0.198264	0.194776	0.192034
5	0.000	0.500	0.217227	0.210696	0.205601
6	0.000	0.625	0.243228	0.232315	0.223884
7	0.000	0.750	0.277698	0.260624	0.247594
8	0.000	0.875	0.322671	0.296998	0.277698
9	0.000	1.000	0.381031	0.343340	0.315505
10	0.125	0.125	0.180369	0.179622	0.179031
11	0.125	0.250	0.188056	0.186129	0.184613
12	0.125	0.375	0.201290	0.197277	0.194139
13	0.125	0.500	0.220746	0.213545	0.207959
14	0.125	0.625	0.247463	0.235660	0.226598
15	0.125	0.750	0.282949	0.264657	0.250791
16	0.125	0.875	0.329365	0.301982	0.281550
17	0.125	1.000	0.389792	0.349639	0.320238
18	0.250	0.250	0.196306	0.193041	0.190493
19	0.250	0.375	0.210538	0.204898	0.200538
20	0.250	0.500	0.231525	0.222241	0.215136
21	0.250	0.625	0.260467	0.245887	0.234866
22	0.250	0.750	0.299129	0.277018	0.260553
23	0.250	0.875	0.350071	0.317297	0.293331
24	0.250	1.000	0.417018	0.369062	0.334754
25	0.375	0.375	0.226542	0.218004	0.211490
26	0.375	0.500	0.250259	0.237240	0.227445
27	0.375	0.625	0.283192	0.263599	0.249091
28	0.375	0.750	0.327589	0.298527	0.277406
29	0.375	0.875	0.386782	0.344098	0.313758
30	0.375	1.000	0.465743	0.403278	0.360043
31	0.500	0.500	0.278208	0.259362	0.245445
32	0.500	0.625	0.317381	0.289880	0.269985
33	0.500	0.750	0.370845	0.330679	0.302296
34	0.500	0.875	0.443261	0.384508	0.344118
35	0.500	1.000	0.541789	0.455399	0.397912
36	0.625	0.625	0.365861	0.326442	0.298650
37	0.625	0.750	0.433053	0.375862	0.336697
38	0.625	0.875	0.525852	0.441980	0.386450
39	0.625	1.000	0.655184	0.530564	0.451255
40	0.750	0.750	0.520868	0.437744	0.382804
41	0.750	0.875	0.644976	0.521917	0.443834
42	0.750	1.000	0.822836	0.636987	0.520868
43	0.875	0.875	0.817853	0.632751	0.520868
44	0.875	1.000	1.073500	0.787799	0.624460
45	1.000	1.000	1.457090	1.004266	0.761476

APPENDIX A(5) IV

No	t	U	VALUES OF INTEGRAL Q		
			a = 1.2 a = 1.4	a = 1.3	
1	0.000	0.000	0.000000	0.000000	0.000000
2	0.000	0.125	0.000000	0.000000	0.000000
3	0.000	0.250	0.000000	0.000000	0.000000
4	0.000	0.375	0.000001	0.000001	0.000001
5	0.000	0.500	0.000003	0.000002	0.000002
6	0.000	0.625	0.000007	0.000005	0.000004
7	0.000	0.750	0.000018	0.000012	0.000008
8	0.000	0.875	0.000046	0.000028	0.000018
9	0.000	1.000	0.000116	0.000066	0.000040
10	0.125	0.125	0.000000	0.000000	0.000000
11	0.125	0.250	0.000000	0.000000	0.000000
12	0.125	0.375	0.000002	0.000001	0.000001
13	0.125	0.500	0.000004	0.000003	0.000002
14	0.125	0.625	0.000011	0.000007	0.000005
15	0.125	0.750	0.000027	0.000017	0.000011
16	0.125	0.875	0.000067	0.000039	0.000024
17	0.125	1.000	0.000169	0.000091	0.000054
18	0.250	0.250	0.000001	0.000001	0.000001
19	0.250	0.375	0.000004	0.000003	0.000002
20	0.250	0.500	0.000007	0.000006	0.000004
21	0.250	0.625	0.000024	0.000015	0.000010
22	0.250	0.750	0.000059	0.000034	0.000021
23	0.250	0.875	0.000150	0.000080	0.000046
24	0.250	1.000	0.000382	0.000187	0.000102
25	0.375	0.375	0.000009	0.000006	0.000004
26	0.375	0.500	0.000023	0.000014	0.000009
27	0.375	0.625	0.000058	0.000033	0.000020
28	0.375	0.750	0.000147	0.000077	0.000045
29	0.375	0.875	0.000374	0.000182	0.000099
30	0.375	1.000	0.000967	0.000434	0.000220
31	0.500	0.500	0.000058	0.000033	0.000020
32	0.500	0.625	0.000146	0.000077	0.000044
33	0.500	0.750	0.000373	0.000182	0.000098
34	0.500	0.875	0.000964	0.000432	0.000218
35	0.500	1.000	0.002527	0.001038	0.000489
36	0.625	0.625	0.000373	0.000181	0.000098
37	0.625	0.750	0.000963	0.000431	0.000218
38	0.625	0.875	0.002526	0.001037	0.000489
39	0.625	1.000	0.006752	0.002526	0.001105
40	0.750	0.750	0.002526	0.001037	0.000489
41	0.750	0.875	0.006752	0.002526	0.001104
42	0.750	1.000	0.018536	0.006255	0.002526
43	0.875	0.875	0.018536	0.006255	0.002526
44	0.875	1.000	0.052851	0.015835	0.005859
45	1.000	1.000	0.159811	0.041325	0.013845

APPENDIX A(5) V

No	t	u	The kernel K(t,u)			
			a = 1.2 a = 5/3	a = 1.3	a = 1.4	
1	0.000	0.000	0.390012	0.426384	0.395928	0.332579
2	0.000	0.125	0.463104	0.427318	0.396676	0.333022
3	0.000	0.250	0.466712	0.430150	0.398939	0.334360
4	0.000	0.375	0.472868	0.434964	0.402776	0.336615
5	0.000	0.500	0.481800	0.441912	0.408291	0.339832
6	0.000	0.625	0.493855	0.451219	0.415634	0.344070
7	0.000	0.750	0.509534	0.463207	0.425022	0.349414
8	0.000	0.875	0.529550	0.478320	0.436743	0.355974
9	0.000	1.000	0.554911	0.497173	0.451192	0.363892
10	0.125	0.125	0.464314	0.428267	0.397433	0.333469
11	0.125	0.250	0.467986	0.431141	0.399726	0.334819
12	0.125	0.375	0.474256	0.436031	0.403615	0.337096
13	0.125	0.500	0.483361	0.443092	0.409205	0.340343
14	0.125	0.625	0.495667	0.452559	0.416656	0.344623
15	0.125	0.750	0.511702	0.464770	0.426189	0.350022
16	0.125	0.875	0.532222	0.480190	0.438107	0.356653
17	0.125	1.000	0.558308	0.499467	0.452819	0.364664
18	0.250	0.250	0.471858	0.434148	0.402109	0.336205
19	0.250	0.375	0.478479	0.439268	0.406155	0.338546
20	0.250	0.500	0.488122	0.446678	0.411980	0.341887
21	0.250	0.625	0.501209	0.456642	0.419760	0.346295
22	0.250	0.750	0.518355	0.469542	0.429741	0.351862
23	0.250	0.875	0.540460	0.485916	0.442265	0.358712
24	0.250	1.000	0.568837	0.506522	0.457798	0.367003
25	0.375	0.375	0.485725	0.444795	0.410474	0.340996
26	0.375	0.500	0.496327	0.452819	0.416709	0.344498
27	0.375	0.625	0.510811	0.463661	0.425065	0.349126
28	0.375	0.750	0.529967	0.477789	0.435836	0.354984
29	0.375	0.875	0.554970	0.495875	0.449432	0.362212
30	0.375	1.000	0.587602	0.518888	0.466426	0.370991
31	0.500	0.500	0.508413	0.461778	0.423560	0.348236
32	0.500	0.625	0.525085	0.473966	0.432786	0.353188
33	0.500	0.750	0.547426	0.489994	0.444756	0.359476
34	0.500	0.875	0.577109	0.510760	0.459997	0.367263
35	0.500	1.000	0.616786	0.537606	0.479257	0.376769
36	0.625	0.625	0.545028	0.488111	0.443251	0.358585
37	0.625	0.750	0.572227	0.506937	0.456947	0.365467
38	0.625	0.875	0.609242	0.531725	0.474581	0.374033
39	0.625	1.000	0.660361	0.564454	0.497191	0.384555
40	0.750	0.750	0.606843	0.529842	0.473075	0.373142
41	0.750	0.875	0.655479	0.560630	0.494140	0.382758
42	0.750	1.000	0.725713	0.602429	0.521657	0.394663
43	0.875	0.875	0.723314	0.600546	0.520151	0.393773
44	0.875	1.000	0.827636	0.656768	0.554947	0.407542
45	1.000	1.000	0.999947	0.737258	0.600874	0.423831

APPENDIX A(5) VI

Integr

al Equation I

No	t	The function g(t)		
		a = 1.2 a = 5/3	a = 1.3	a = 1.4
1	0.000	2.06544	0.86815	1.73636
2	0.125	1.53156		
3	0.250	2.07518	1.87072	1.73815
4	0.375	1.53238		
5	0.500	2.08731	1.87856	1.74360
6	0.625	1.53493		
7	0.750	2.10840	1.98204	1.75293
8	0.875	1.53924		
9	1.000	2.13994	1.91189	1.76651
		1.54543		
		2.18443	1.93926	1.78496
		1.55365		
		2.25301	1.97595	1.80918
		1.56415		
		2.33217	2.02478	1.84052
		1.57722		
		2.45673	2.09044	1.88103
		1.59328		

Appendix A(5) VII

$$\int_{x_0}^{x_4} f(x) dx = \frac{2h}{45} [7f_0 + 32f_1 + 12f_2 + 32f_3 + 7f_4] = -\frac{8h^7}{945} f^{vi}(\xi)$$

$$\int_{x_0}^{x_5} f(x) dx = \frac{5h}{288} [19f_0 + 75f_1 + 50f_2 + 50f_3 + 75f_4 + 19f_5] - \frac{275h^7}{12096} f^{vi}(\xi)$$

$$\int_{x_0}^{x_6} f(x) dx = \frac{h}{140} [41f_0 + 216f_1 + 27f_2 + 272f_3 + 27f_4 + 216f_5 + 41f_6] - \frac{9h^9}{1400} f^{viii}(\xi)$$

$$\int_{x_0}^{x_7} f(x) dx = \frac{7h}{17280} [751f_0 + 3577f_1 + 1323f_2 + 2989f_3 + 2989f_4 + 1323f_5 + 3577f_6 + 751f_7]$$

$$-\frac{8183h^9}{518400} f^{viii}(\xi)$$

APPENDIX A(5) VIII

			<u>Integral</u> <u>Equation II</u>				
u	t	A	B Kernel	C	D	E	
0	0	0.00000 0	0.00000 0	0.00000 0	0.00000 0	0.00000 0	0.00000 0
.1	0	0.00000 0	0.00000 0	0.00000 0	0.00000 0	0.00000 0	0.00000 0
.1	.1	0.01170 2	- 0.000032	0.07004 2	0.0560 12	0.00026 6	- 0.957328
.1	.2	0.02195 3	- 0.000061	0.14008 4	0.1134 45	0.00059 1	- 1.915845
.1	.3	0.02930 0	- 0.000082	0.21012 7	0.1737 79	0.00104 5	- 2.876770
.1	.4	0.03228 0	- 0.000090	0.28016 9	0.2386 18	0.00173 0	- 3.841391
.1	.5	0.02415 0	- 0.000082	0.35021 1	0.3097 58	0.00280 7	- 4.811098
.1	.6	0.01921 0	- 0.000054	0.42025 3	0.3892 73	0.00453 2	- 5.787446
.1	.7	0.00014 8	- 0.000001	0.49029 6	0.4796 22	0.00732 8	- 6.772218
.2	0	0.00000 0	- 0.000000	0.00000 0	0.0000 00	0.00000 0	0.000000
.2	.2	0.04100 2	- 0.000114	0.28016 9	0.2297 91	0.00131 1	- 3.834098
.2	.3	0.05423 3	- 0.000151	0.42025 3	0.3520 63	0.00232 1	- 5.757235
.2	.4	0.05871 5	- 0.000164	0.56033 8	0.4835 37	0.00385 2	- 7.687868
.2	.5	0.05149 0	- 0.000144	0.70042 2	0.6278 91	0.00626 2	- 9.628836
.2	.6	0.02956 3	- 0.000083	0.84050 7	0.7893 80	0.01013 6	- 11.58331 6
.2	.7	0.01010 6	0.00002 8	0.98059 1	0.97305 3	0.01644 5	- 13.55098 6
.3	0	0.00000 0	0.00000 0	0.00000 0	0.00000 0	0.00000 0	0.000000

.3	.	0.07041	-	0.63038	0.5395	0.00411	-8.645196
	3	7	0.000196	0	49	8	
.3	.	0.07344	-	0.84050	0.7413	0.00685	-
	4	4	0.000205	7	36	3	11.544681
.3	.	0.05886	-	1.05063	0.9631	0.01118	-
	5	4	0.000164	3	59	0	14.460086
.3	.	0.02217	-	1.26076	1.2116	0.01817	-
	6	4	0.000062	0	71	6	17.39637
							6
.3	.	-	0.00011	1.47088	1.49478	0.02964	-
	7	0.04119	5	7	6	6	20.35934
		9					8
.4	0	0.00000	0.00000	0.00000	0.00000	0.00000	0.000000
		0	0	0	0	0	
.4	.	0.07056	-	1.12067	1.0191	0.01144	-
	4	5	0.000197	6	71	7	15.41741
							5
.4	.	0.04412	-	1.40084	1.3251	0.0187	-
	5	7	0.000123	5	16	66	19.312221
.4	.	-	0.00003	1.68101	1.66856	0.03069	-
	6	0.01189	3	3	5	1	23.236118
		9					
.4	.	-	0.00029	1.96118	2.06074	0.05043	-
	7	0.10363	0	3	2	1	27.197445
		2					
.5	0	0.00000	0.00000	0.00000	0.00000	0.00000	0.000000
		0	0	0	0	0	
.5	.	-	0.00000	1.75105	1.72457	0.03095	-
	5	0.000197	1	6	7	7	24.19344
							6
.5	.	-	0.00022	2.10126	2.17418	0.05102	-
	6	0.081679	9	7	7	2	29.11328
							9
.5	.	-	0.00058	2.45147	2.68912	0.08465	-
	7	0.208005	4	8	2	5	34.08333
							3
.6	0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000
		0	0	0	0	0	0
.6	.	-	0.00055	2.52152	2.74513	0.08492	-
	6	0.196304	2	1	4	2	35.040661
.6	.	-	0.00102	2.94177	3.40144	0.14268	-
	7	0.365052	7	4	1	4	41.034296
.7	0	0.00000	0.00000	0.00000	0.00000	0.00000	0.00000

		0	0	0	0	0	0
.7	.	-	0.00165	3.43207	4.22383	0.24374	-
	7	0.585672	1	0	5	0	48.073148

APPENDIX A(5) IX

Equation II		Integral		
No. c =0.7	g(t)	c = 0.4	c = 0.5	c =0.6
1 0	g(0.0)	0	0	0
2 0.00892	g(0.1)	0.03291	0.02039	0.01299
3 0.01771	g(0.2)	0.06574	0.03488	0.02586
4 0.02624	g(0.3)	0.09839	0.06075	0.03850
5 0.03437	g(0.4)	0.13078	0.08049	0.05076
6 0.04192	g(0.5)		0.09978	0.06250
7 0.04872	g(0.6)			0.07356
8 0.05452	g(0.7)			

APPENDIX A(5) X

		Integral Equation II	
Function c =0.7	c = 0.4	c = 0.5	c = 0.6
$\int_0^t g(t)dt$	0.00700	0.00829	0.00905

0.00946

→ESTABLISH DBD002A00KP4;
SOLVES POISSON S EQUATION;
O/P 8;→

```

begin library A0,A6,A7,A8,A9,A2700;
  comment Here the user is reqd to feed in the 5 procedures POISSON,DIRI,DESCRT,DERV,NEUM.
    It could be done in any order;

  procedure POISSON(x,y,r);value x,y; real x,y,r; r:=0;

  procedure DIRI(x,y,z,d);value x,y,z; real x,y,d; integer z;
    begin if z=3 then d:=x2+y2;if z=2 then d:=y; end;

  procedure DESCRT(X,Y,z,x,y);value X,Y,z; real X,Y,x,y; integer z;
    begin if z=1 then begin x:=sqrt(1.0-(Y2)); y:=sqrt(1.0-(X2)); end;
      if z=2 then begin x:=0;y:=Y;end;
      if z=3 then begin x:=X;y:=0;end;
    end;

  procedure DERV(x,y,z,a);value x,y,z; real x,y,a; integer z;
    begin if z=1 then a:=3.14159265+arctan(-x/y);
      if z=2 then a:=3.14159265/2;if z=3 then a:=0;
    end;

  procedure NEUM(x,y,z,d);value x,y,z; real x,y,d; integer z;
    begin if z=1 then d:=y;end;

  real h,x,y,X,Y,RHS,D1,h1,h2,m1,m2,d,A,N1,N2;
  integer f,w,ff,cols,i,nsor,ax,ay,k,pt,band,b,Z,acont,s,node,type,cut,totala,
    sgn,zero,n,ub,natt,atype,q,a12,a21,a43,a34,a58,a85,a67,a76,a912,a129,
    a1011,a1110,sw,uw,bd,SP1,SO,SM1,SM2,UP1,UO,UM2,UM1,TO,ffail,sp,sm,b1,b2,
    b3,b4,b5,b6,b7,b8,b9,b10;
  integer array B,T,M[-2:2],a,P[0:12];
  array V[0:12];

  open(20); f:=format([-ndddd;]);find(100,[DGO10002]); find(101,[DGO10003]);
  interchange(100); interchange(101);
  write text(0,[[cc]PART*I[cc]]);

  acont:=read(20); node:=read(20);
  b:=read(20);zero:=read(20);sgn:=read(20);ax:=read(20);ay:=read(20);
  h:=read(20);ub:=read(20);totala:=read(20);cols:=read(20);ff:=read(20);

  a21:=2xax+ay; a43:=4xax+3xay; a58:=5xax+8xay; a67:=6xax+7xay;
  a912:=9xax+12xay; a1011:=10xax+11xay;a12:=ax+2xay; a34:=3xax+4xay;
  a85:=8xax+5xay; a76:=7xax+6xay; a129:=12xax+9xay; a1110:=11xax+10xay;
  sp:=(1+sgn)/2; sm:=(1-sgn)/2;

```

```

b1:=spxa129+smxa85; b2:=spxa1110 + smxa76;b3:=spxa58 + smxa912;
b4:=spxa21 + smxa43; b5:=spxa1011+smxa67; b6:=spxa912 + smxa1011;
b7:=spxa43 + smxa21; b8:=spxa67 + smxa58;
b9:=spxa85 + smxa1110;b10:=spxa76 + smxa129;

```

```

begin integer array NODE,TYPE[1:nodex6],ORDNT[1:acnt+2],CUT[1:nodex20],R[1:ub],Eq[0:12];
array BAND[1:4xb+4],COUNT[1:9],t[1:1],SIZE,pp[1:2],Q[1:zero];

```

```

procedure EQ(p,k,Z); value p,k,Z; integer p,k,Z;
comment form the band equation procedure;
begin if p=1 then bd:=2xb+2+B[0]-band+Eq[k]+B[-sgn];
if p=2 then bd:=2xb+2+B[0]-band+Eq[k];
if p=3 then bd:=2xb+2-band-B[sgn]+Eq[k];
if p=4 then bd:=2xb+2-band-B[sgn]-B[2xsgn]+Eq[k];

if P[k]≠0 then begin if TYPE[Z]≠0 then BAND[bd]:=V[k]; end;
end;

```

```

procedure Find(nm,pq,rs,I); value I; integer nm,pq,rs,I;
comment finds 3 parameters reqd through out the prog;
begin integer jj,kk;
if NODE[sw]+nm=NODE[I+pq] then
begin P[rs]:=I;

for kk:=1 step 1 until I do
if TYPE[kk+pq]≠0 then Eq[rs]:=Eq[rs]+1;

for jj:=1 step 1 until (I-1) do
begin for kk:=0,-1,-2,-3,-4,-11,-22,-33,-44 do
if TYPE[jj+pq]=kk then a[rs]:=a[rs]+1;

for kk:=-12,-13,-14,-23,-24,-34,-1122,-2233,-3344,-1144 do
if TYPE[jj+pq]=kk then a[rs]:=a[rs]+2;
end;
a[rs]:=a[rs]+1;
end;
end;

```

```

procedure ARM(p1,p2,pp); integer p1,p2,pp;
comment finds the coeff and RHS if any of the 4 neighbouring nodes lies on
Dirichlet boundary;
begin array D[1:4];
if TYPE[p1]=0 then begin DESCRT(X,Y,p2,x,y);
if pp=1 or pp=3 then DIRI(x,Y,p2,D[pp]);
if pp=2 or pp=4 then DIRI(X,y,p2,D[pp]);
RHS:=RHS-V[pp]xD[pp]; V[pp]:=0;
end;
end;

```

```

procedure ZERO(t1,t2,t3,t4); integer t1,t2,t3,t4;
  begin
    if t3 $\neq$ 0 then ARM(axX(sw-1)+ayX(SO+P[3]),axXCUT[UM1+a[3]]+ayXCUT[UO+a[3]],3);
    if t2 $\neq$ 0 then ARM(axX(SM2+P[2])+ayX(sw+1),axXCUT[UM2+a[2]]+ayXCUT[UM1+a[2]],2);
    if t4 $\neq$ 0 then ARM(axX(SO+P[4])+ayX(sw-1),axXCUT[UO+a[4]]+ayXCUT[UM1+a[4]],4);
    if t1 $\neq$ 0 then ARM(axX(sw+1)+ayX(SM2+P[1]),axXCUT[UM1+a[1]]+ayXCUT[UM2+a[1]],1);
  end;

```

```

real procedure tan(a); value a; real a; tan:=sin(a)/cos(a);

```

```

procedure STAR(n,m); integer n,m;
  comment calcs coeffs for the 5-point star node;
  begin real h1,h2,h3,h4; real array D[1:4];
    h1:=h2:=h3:=h4:=h; ffail:=10;

    if n>0 then DESCRT(X,Y,CUT[uw],x,y);
    if n=1 then begin h1:=abs(x-X); DIRI(x,Y,CUT[uw],D[1]); end;
    if n=2 then begin h2:=abs(y-Y); DIRI(X,y,CUT[uw],D[2]); end;
    if n=3 then begin h3:=abs(x-X); DIRI(x,Y,CUT[uw],D[3]); end;
    if n=4 then begin h4:=abs(y-Y); DIRI(X,y,CUT[uw],D[4]); end;

    if m>0 then DESCRT(X,Y,CUT[uw+1],x,y);
    if m=2 then begin h2:=abs(y-Y); DIRI(X,y,CUT[uw+1],D[2]); end;
    if m=3 then begin h3:=abs(x-X); DIRI(x,Y,CUT[uw+1],D[3]); end;
    if m=4 then begin h4:=abs(y-Y); DIRI(X,y,CUT[uw+1],D[4]); end;

    if h1>h or h2>h or h3>h or h4>h then goto FF;

    V[0]:=-2XhXhX((1/(h1Xh3)) + (1/(h2Xh3)));
    V[1]:=(2XhXh)/(h1X(h1+h3)); V[2]:=(2XhXh)/(h2X(h2+h4));
    V[3]:=(2XhXh)/(h3X(h3+h1)); V[4]:=(2XhXh)/(h4X(h4+h2));

    if n>0 then begin RHS:=RHS-V[n]XD[n]; V[n]:=0; end;
    if m>0 then begin RHS:=RHS-V[m]XD[m]; V[m]:=0; end;

    ZERO((1-n)X(1-m),(2-n)X(2-m),(3-n)X(3-m),(4-n)X(4-m));
  end;

```

```

procedure ONE(aa,bb,cc,dd); integer aa,bb,cc,dd;
comment calcs coeffs when one arm is cut by Neumann bounrary;
begin real P5,Q5,R5,S5; integer tt2,tt4,st1;

Z:=CUT[uw];DESCRT(X,Y,Z,x,y);ffail:=1;
if aa=1 or cc=1 then
begin DERV(x,Y,Z,A);NEUM(x,Y,Z,N1);h1:=abs(x-X);
    if abs(A)=0 or abs(A)=3.14159265 then goto FF;
    d:=tan(A-1.570796325); m1:=abs(d);
end;
if bb=1 or dd=1 then
begin DERV(X,y,Z,A);NEUM(X,y,Z,N1);h1:=abs(y-Y);
    if abs(A)=1.570796325 or abs(A)=4.712388975 then goto FF;
    d:=tan(A); m1:=abs(d);
end;

V[0]:=-4x(h+h1);P5:=h+h1+h1-(m1Xh);Q5:=2x(h-(m1Xh1));
R5:=m1x(h+h1)+h+h1+h1; S5:=m1Xh1; ffail:=2;
RHS:=RHSx(h+h1+h1+(m1Xh1))-2XhXhXN1X(sqrt(1.0+(m1Xm1)));
if abs(V[0])<(P5+Q5+R5+S5-0.05) then goto FF;

V[aa+2Xbb+3Xcc+4Xdd]:=0;    tt2:=2Xaa+3Xbb+4Xcc+dd;
V[3Xaa+4Xbb+cc+2Xdd]:=Q5;   tt4:=4Xaa+bb+2Xcc+3Xdd;

if d<0 then begin st1:=6Xaa+8Xbb+10Xcc+12Xdd; ffail:=3;
    if P[st1]=0 then goto FF;
    V[st1]:=S5; V[tt2]:=R5; V[tt4]:=P5;

    Z:=CUT[aaX(UOXax+UP1Xay)+bbX(UP1Xax+UM2Xay)+ccXaxXUM2+ddXayXUO+a[st1]];
    if TYPE[aaX(SOXax+SP1Xay)+bbX(SP1Xax+SM2Xay)+ccXaxXSM2+ddXayXSO+P[st1]]=0 then
    begin if aa=1 then DIRI(X-2Xh,Y-h,Z,D1);
        if bb=1 then DIRI(X+h,Y-2Xh,Z,D1);
        if cc=1 then DIRI(X+2Xh,Y+h,Z,D1);
        if dd=1 then DIRI(X-h,Y+2Xh,Z,D1);
        RHS:=RHS-V[st1]XD1; V[st1]:=0;
    end;
end;

if d>0 then begin st1:=5Xaa+7Xbb+9Xcc+11Xdd; ffail:=4;
    if P[st1]=0 then goto FF;
    V[st1]:=S5; V[tt2]:=P5; V[tt4]:=R5;

    Z:=CUT[aaX(axXUM2+ayXUP1)+bbX(axXUP1+ayXUO)+ccXaxXUO+ddXayXUM2+a[st1]];
    if TYPE[aaX(axXSM2+ayXSP1)+bbX(axXSP1+ayXSO)+ccXaxXSO+ddXayXSM2+P[st1]]=0 then
    begin if aa=1 then DIRI(X-2Xh,Y+h,Z,D1);
        if bb=1 then DIRI(X-h,Y-2Xh,Z,D1);
        if cc=1 then DIRI(X+2Xh,Y-h,Z,D1);
        if dd=1 then DIRI(X+h,Y+2Xh,Z,D1);
    end;

```

```

RHS:=RHS-V[st1]XD1; V[st1]:=0;
end;
end;
ZERO(1-aa,1-bb,1-cc,1-dd);
end;

procedure TWO(aa,bb,cc,dd); integer aa,bb,cc,dd;
comment calcs coeffs when two arms are cut by Neumann boundary;
begin real x1,x2,y1,y2,A1,A2,EX5,EX11,EX12,EX21,EX22,E1,E2,E3,E4;
integer st1,st2,st3,st4,st5,st6,st7,st8,Z1,Z2;
st1:=5Xaa+7Xbb+9Xcc+11Xdd; st3:=3Xaa+4Xbb+cc+2Xdd;
st2:=8Xaa+10Xbb+12Xcc+6Xdd; st4:=4Xaa+bb+2Xcc+3Xdd;

ffail:=5;
if P[st1]=0 and P[st2]=0 then goto FF;
Z1:=CUT[uw]; DESCRT(X,Y,Z1,x1,y1);
Z2:=CUT[uw+1]; DESCRT(X,Y,Z2,x2,y2);

ffail:=6;
if aa=1 or cc=1 then
begin DERV(x1,Y,Z1,A1); NEUM(x1,Y,Z1,N1); h1:=abs(x1-X);
if abs(A1)=0 or abs(A1)=3.14159265 then goto FF;
m1:=tan(A1-1.570796325);
DERV(X,y2,Z2,A2); NEUM(X,y2,Z2,N2); h2:=abs(y2-Y);
if abs(A2)=1.570796325 or abs(A2)=4.712388975 then goto FF;
m2:=tan(A2);
end;
if bb=1 or dd=1 then
begin DERV(X,y1,Z1,A1); NEUM(X,y1,Z1,N1); h1:=abs(y1-Y);
if abs(A1)=1.570796325 or abs(A1)=4.712388975 then goto FF;
m1:=tan(A1);
DERV(x2,Y,Z2,A2); NEUM(x2,Y,Z2,N2); h2:=abs(x2-X);
if abs(A2)=0 or abs(A2)=3.14159265 then goto FF;
m2:=tan(A2-1.570796325);
end;

V[aa+2Xbb+3Xcc+4Xdd]:=V[2Xaa+3Xbb+4Xcc+dd]:=0;
V[0]:=-2XhX(h+h1+h2+(m1Xh2)-(m2Xh1)+(m1Xm2Xh));

RHS:=RHSX((h/2)X(hX(1.0+m1Xm2)+(m1Xh1)-(m2Xh2)+(m1Xm2X(h1+h2)))+(hX(h1+h2))+(h1Xh2X(2+m1-m2)));
RHS:=RHS-hXhX((N1X(sqrt(1.0+(m1Xm1)))X(h+h2+h2+m2Xh))+N2X(sqrt(1.0+(m2Xm2)))X(h+h1+h1-(m1Xh)));
EX5:=(h1Xh2X(m1-m2))+(h/2)X((m1Xh1)-(m2Xh2)+(m1Xm2X(h1+h2)));

E1:=(hXhX(1.0+(m1Xm2)))+(2XhX(h2-(h1Xm2)));
E2:=(m1Xm2XhX(h1+h2))+(hX(m1Xh1-m2Xh2))+(2Xh1Xh2X(m1-m2));
E3:=(hXhX(1.0+(m1Xm2)))+(2XhX(h1+m1Xh2));
E4:=(h1Xh2X(m1-m2))+(h/2)X((m1Xh1)-(m2Xh2)+(m1Xm2X(h1+h2)));

EX11:=E1-E2; EX12:=E3+E4;

```

if abs(V[0])<(EX5+EX11+EX12-0.05) then goto J1;

if P[st1]≠0 then

begin V[st1]:=EX5; V[st3]:=EX11; V[st4]:=EX12;

st5:=aa×SM2+bb×SP1+cc×SO+P[st1]; st7:=aa×SP1+bb×S+dd×SM2+P[st1];

st6:=aa×UM2+bb×UP1+cc×UO+a[st1]; st8:=aa×UP1+bb×UO+dd×UM2+a[st1];

Z:=CUT[ax×st6+ay×st8];

if TYPE[ax×st5+ay×st7]=0 then

begin if aa=1 then DIRI(X-2×h,Y+h,Z,D1);

if bb=1 then DIRI(X-h,Y-2×h,Z,D1);

if cc=1 then DIRI(X+2×h,Y-h,Z,D1);

if dd=1 then DIRI(X+h,Y+2×h,Z,D1);

RHS:=RHS-V[st1]×D1; V[st1]:=0;

end;

goto J2;

end;

J1: EX21:=E1+E4; EX22:=E3-E2; ffail:=9;

if abs(V[0])<(EX21+EX22+EX5-0.05) then goto FF;

V[st2]:=EX5; V[st3]:=EX21; V[st4]:=EX22;

st5:=aa×SP1+bb×SM2+dd×SO+P[st2]; st7:=aa×SM2+cc×SO+dd×SP1+P[st2];

st6:=aa×UP1+bb×UM2+dd×UO+a[st2]; st8:=aa×UM2+cc×UO+dd×UP1+a[st2];

Z:=CUT[ax×st6+ay×st8];

if TYPE[ax×st5+ay×st7]=0 then

begin if aa=1 then DIRI(X+h,Y-2×h,Z,D1);

if bb=1 then DIRI(X+2×h,Y+h,Z,D1);

if cc=1 then DIRI(X-h,Y+2×h,Z,D1);

if dd=1 then DIRI(X-2×h,Y-h,Z,D1);

RHS:=RHS-V[st2]×D1; V[st2]:=0;

end;

J2: ZERO((1-aa)×(1-dd),(1-aa)×(1-bb),(1-bb)×(1-cc),(1-cc)×(1-dd));

end;

T[sgn]:=M[sgn]:=T[2Xsgn]:=M[2Xsgn]:=s:=node:=type:=pt:=cut:=acont:=q:=nsor:=0;

READ:

atype:=read(20);
if atype#1 then natt:=0 else natt:=-1;

ub:=read(20);
for i:=1 step 1 until ub do
R[i]:=read(20);

totala:=totala-1;

if s=0 then goto G;

COPY:

if sgn=1 then begin n:=0;
 for i:=1 step 1 until R[2] do
 for k:=R[2Xi+1] step 1 until R[2Xi+2] do
 n:=n+1;
 if s=-1 or s=-2 then
 begin if s=-1 then bd:=T[0];
 if s=-2 then bd:=T[-1]+T[0]+T[1]+T[2];
 for i:=1 step 1 until bd do
 begin NODE[n+1+bd-1]:=NODE[bd+1-1];
 TYPE[n+1+bd-1]:=TYPE[bd+1-1];
 end;
 end;
 end;
 node:=type:=0;
 end;

G:

n:=0;

for i:=1 step 1 until R[2] do
for k:=R[2Xi+1] step 1 until R[2Xi+2] do
begin node:=node+1; NODE[node]:=k; n:=n+1; end;

T[s]:=n;
for i:=1 step 1 until n do
begin if R[2XR[2]+2+i]<0 then begin type:=type+1; TYPE[type]:=R[2XR[2]+2+i]; goto L; end;
 for k:=1 step 1 until R[2XR[2]+2+i] do
 begin type:=type+1; TYPE[type]:=1; end;
 n:=n+1-R[2XR[2]+2+i];
end;

L:

end;

M[s]:=ub-2X(R[2]+1)-n;

if sgn=1 then begin if s=-1 then begin TO:=M[0];
 for i:=1 step 1 until TO do
 CUT[M[-1]+TO+1-1]:=CUT[TO+1-1]; end;
 if s=-2 then begin bd:=M[-1]+M[0]+M[1]+M[2];

```

for i:=1 step 1 until bd do
  CUT[bd+M[-2]+1-i]:=CUT[bd+1-i];

```

```

end;

```

```

cut:=0;

```

```

end;

```

```

for i:=1 step 1 until M[s] do
  begin cut:=cut+1; CUT[cut]:=R[2+2×R[2]+n+1]; end;

```

C:

```

acont:=acont+1;
if natt>0 then begin ORDNT[acont]:=R[1]+(natt×sgn); atype:=atype-1; natt:=natt+1; end;
if natt=-1 then begin ORDNT[acont]:=R[1]; atype:=0; end;

```

```

if s×sgn≠-2 then begin s:=s-sgn; if atype=0 then goto READ else goto COPY ; end;

```

```

SM2:=T[-2]; SM1:=SM2+T[-1]; SO:=SM1+T[0]; SP1:=SO+T[1];
UM2:=M[-2]; UM1:=UM2+M[-1]; UO:=UM1+M[0]; UP1:=UO+M[1];

```

```

TO:=T[0];

```

```

t[1]:=T[0]; write binary(101,t,[t]);
for i:=1 step 1 until TO do
  begin pp[1]:=NODE[SM1+i]; pp[2]:=TYPE[SM1+i]; write binary(101,pp,[PAIR]); end;

```

```

nsor:=nsor+TO;

```

```

w:=0;
for i:=1 step 1 until 5 do
  begin B[i-3]:=0; for k:=1 step 1 until T[i-3] do
    begin w:=w+1; if TYPE[w]≠0 then B[i-3]:=B[i-3]+1; end;
  end;

```

```

band:=0;

```

```

comment the following w loop scans the nodes of the row, it finishes with setting up
band eq in the coeff matrix;

```

```

for w:=1 step 1 until TO do

```

```

  begin sw:=SM1+w;

```

```

    if ax=1 then begin X:=NODE[sw]×h; Y:=ORDNT[acont-2]×h; end;
    if ay=1 then begin Y:=NODE[sw]×h; X:=ORDNT[acont-2]×h; end;
    POISSON(X,Y,RHS); RHS:=h×h×RHS;

```

```

    for i:=0 step 1 until 12 do

```



```

begin Eq[i]:=a[i]:=P[i]:=0;
      V[i]:=0;
end;

for i:=1 step 1 until SM2 do
begin Find(-1,0,a129,i); Find(1,0,a1110,i); end;

for i:=1 step 1 until T[-1] do
begin Find(-2,SM2,a58,i); Find(0,SM2,a21,i); Find(2,SM2,a1011,i); end;

for i:=1 step 1 until T[1] do
begin Find(-2,SO,a67,i); Find(0,SO,a43,i); Find(2,SO,a912,i); end;

for i:=1 step 1 until T[2] do
begin Find(-1,SP1,a76,i); Find(1,SP1,a85,i); end;

if w>1 then Find(-1,SM1,a34,w-1);

Find(0,SM1,0,w); uw:=UM1+a[0];

if w≠TO then Find(1,SM1,a12,w+1);

if TYPE[sw]=0 then begin q:=q+1; DIRI(X,Y,CUT[uw],Q[q]); goto ZZ; end;
if TYPE[sw]=-1 then STAR(1,0);
if TYPE[sw]=-2 then STAR(2,0);
if TYPE[sw]=-3 then STAR(3,0);
if TYPE[sw]=-4 then STAR(4,0);
if TYPE[sw]=-12 then STAR(1,2);
if TYPE[sw]=-13 then STAR(1,3);
if TYPE[sw]=-14 then STAR(1,4);
if TYPE[sw]=-23 then STAR(2,3);
if TYPE[sw]=-24 then STAR(2,4);
if TYPE[sw]=-34 then STAR(3,4);
if TYPE[sw]=1 then STAR(0,0);
if TYPE[sw]=-11 then ONE(1,0,0,0);
if TYPE[sw]=-22 then ONE(0,1,0,0);
if TYPE[sw]=-33 then ONE(0,0,1,0);
if TYPE[sw]=-44 then ONE(0,0,0,1);
if TYPE[sw]=-1122 then TWO(1,0,0,0);
if TYPE[sw]=-2233 then TWO(0,1,0,0);
if TYPE[sw]=-3344 then TWO(0,0,1,0);
if TYPE[sw]=-1144 then TWO(0,0,0,1);

pt:=pt+1; band:=band+1;
for i:=1 step 1 until 4xb+3 do
BAND[i]:=0;
N1:=BAND[2xb+2]:=V[0];
if P[a12]≠0 then begin if TYPE[SM1+P[a12]]≠0 then BAND[2xb+3]:=V[a12] end;
if P[a34]≠0 then begin if TYPE[SM1+P[a34]]≠0 then BAND[2xb+1]:=V[a34] end;

```

```

EQ(1,b1,spXa129+smX(SP1+P[a85]));
EQ(1,b2,spXa1110 + smX(SP1+P[a76]));
EQ(2,b3,spX(SM2+P[a58]) + smX(SO+P[a912]));
EQ(2,b4,spX(SM2+P[a21]) + smX(SO+P[a43]));
EQ(2,b5,spX(SM2+P[a1011])+smX(SO+P[a67]));
EQ(3,b6,spX(SO+P[a912]) + smX(SM2+P[a1011]));
EQ(3,b7,spX(SO+P[a43]) + smX(SM2+P[a21]));
EQ(3,b8,spX(SO+P[a67]) + smX(SM2+P[a58]));
EQ(4,b9,spX(SP1+P[a85])+smXP[a1110]);
EQ(4,b10,spX(SP1+P[a76])+smXP[a129]);

```

BAND[4xb+4]:=RHS;

for i:=1 step 1 until (4xb+4) do
 BAND[i]:=BAND[i]/N1;

write binary(100,BAND,[BAND]);

ZZ: end;

if sgn=1 then begin for i:=4 step -1 until 1 do
 begin T[i-2]:=T[i-3]; M[i-2]:=M[i-3]; end;
 goto GG;

end;
 bd:=T[2]+T[1]+T[0]+T[-1];

for i:=1 step 1 until bd do
begin NODE[i]:=NODE[T[-2]+i]; TYPE[i]:=TYPE[T[-2]+i]; end;

bd:=M[2]+M[1]+M[0]+M[-1];
for i:=1 step 1 until bd do
 CUT[i]:=CUT[M[-2]+i];

for i:=1 step 1 until 4 do
begin T[i-3]:=T[i-2]; M[i-3]:=M[i-2]; end;

type:=node:=T[-2]+T[-1]+T[0]+T[1];
 cut:=M[-2]+M[-1]+M[0]+M[1];

GG:

if totala=-2 then goto F;
if totala=-1 then begin totala:=-2; T[-sgn]:=M[-sgn]:=0; goto C; end;
if totala= 0 then begin if atype=0 then begin totala:=-1;
 T[-2xsgn]:=M[-2xsgn]:=0;
 if natt=-1 then natt:=1; goto C; end;
 if atype>0 then goto COPY;
end;
if totala> 0 then begin if atype=0 then goto READ;
 if atype>0 then goto COPY; end;

F:

if cols>8 then cols:=8;
 COUNT[1]:=nsor;
 COUNT[2]:=acont-2;

```

COUNT[3]:=sgn;
COUNT[4]:=pt;
COUNT[5]:=cols;
COUNT[6]:=ff;
COUNT[7]:=q;
COUNT[8]:=ax;
COUNT[9]:=ay;
write binary(101,COUNT,[COUNT]);
begin array QQ[1:q];
  for i:=1 step 1 until q do
    QQ[i]:=Q[i];

    write binary(101,QQ,[ZEROS]);
  end;
close(101);
SIZE[1]:=pt; SIZE[2]:=4xb+3; write binary(100,SIZE,[SIZE]);
close(100);close(20);

ENTER([DBD002BOOKP4]);
end;
FF: write text(0,[c]FAULT*IN*DATA*AT*NODE*);write(0,f,w); write(0,f,acont-2);
write text(0,[c]FAILURE*NUMBER*); write(0,f,ffail); close(20);close(100);close(101);

write text(0,[cc]TERMINATE * PROGRAM);
end→

→ESTABLISH DBD002BOOKP4;
TRIM PROGRAM PART II;
O/P 8;→

begin library A0,A6,A7,A8,A9,A2700;
integer i,j,k,n,band,f;
find(100,[DGO10002]); f:=format([s-ndddd;d;c]);
write text(0,[ccc]PART*II[cc]);
begin array SIZE[1:2];
  read binary(100,SIZE,[SIZE]);
  n:=SIZE[1]; band:=SIZE[2];

  k:=n+1; skip(100,-k);
  begin array a[1:band+1];
    j:=-1;
    TR: j:=j+1;
    for i:=1 step 1 until n do
      begin read binary(100,a,[BAND]);
        if a[j+1]≠0 or a[band-j]≠0 then goto J;
      end;
      rewind(100); goto TR;
    end;
    J: write text(0,[matrix*trimmed[cc]]); rewind(100);
  end;
end;

```

```

SIZE[1]:=n; SIZE[2]:=((band-j-j)+2);

write text(0,[number*of*rows*is**]); write(0,f,n);
write text(0,[half*band*width*is*]); write(0,f,SIZE[2]);

begin array A[1:n , (-SIZE[2]):(SIZE[2])],B[1:n];
  for i:=1 step 1 until n do
    begin read binary(100,a,[BAND]);
      for k:=1 step 1 until (band-j-j) do
        A[i,k-1-SIZE[2]]:=a[j+k];

        B[i]:=a[band+1];
      end;

      rewind(100); interchange(100);
      write binary(100,SIZE,[SIZE]);
      write binary(100,A,[A]);
      write binary(100,B,[B]);
    end;
  end;
end;
close(100);
write text(0,[PART*II*OVER[cc]]);
ENTER([DBD002C--KP4]);
end→

→ESTABLISH DBD002COOKP4;
SOLVE PROGRAM PART III;
O/P 8;→

begin library AO,A6,A7,A8,A9,A2700;
  integer i,j,k,n,w;
  find(100,[DGO10002]); write text(0,[[cc]PART**III[cc]]);
  begin array S[1:2]; read binary(100,S,[SIZE]); n:=S[1]; w:=S[2];
    begin array A[1:n , -w:w]; read binary(100,A,[A]);

      for j:=1 step 1 until n do
        begin for i:= (if j>w then j-w+1 else 2 ) step 1 until (if j<n-w then j+w else n) do
          for k:=if j>1 and j>w then j-w else if not j>1 and i>w then i-w else
            1 step 1 until (if j>1 then i-1 else j-1) do
            A[i,j-1]:=A[i,j-1]-A[i,k-1]*A[k,j-k];

            if A[j,0]=0.0 then begin write text(0,[diagonal*element*zero*TERMINATE*PROG]); goto Z;end;

            for i:=j+1 step 1 until (if j<n-w then j+w else n) do
              A[i,j-1]:=A[i,j-1]/A[j,0];
          end;

          write text(0,[[c]triangular*decomposition*over[c]]);

```

```

begin array B[1:n]; read binary(100,B,[B]);

for i:=2 step 1 until n do
for j:=1 step 1 until (if i<w then i-1 else w) do
B[i]:=B[i]-A[i,if i<w+1 then j-1 else j-w-1]*B[if i<w+1 then j else j+i-w-1];

for i:=n step -1 until 1 do
begin A[1,-w]:=0;
for j:=1 step 1 until (if i>n-w+1 then n-1 else w) do
A[1,-w]:=A[1,-w]+A[i,j]*B[i+j];

B[i:]=(B[i]-A[1,-w])/A[i,0];

end;

write text(0,[[c]solution*obtained[c]]);
rewind(100);interchange(100);
write binary(100,B,[RESULT]);
end;

end;

end;
write text(0,[PART*III*OVER]);
close(100);
ENTER([DBD002D--KP4]);
Z: close(100);
end→

```

→ESTABLISH DBD002DOOKP4;
OUTPUT PROGRAM PART IV;
O/P 8;→

```

begin library A0,A6,A7,A8,A9,A12,A2700;
integer nsor,acont,sgn,k,i,j,pt,n,min,f,oldnode,newnode,oldrslt,newrslt,cols,zero,
oldzero,newzero,sum,ax,ay,v,m;
real r;
array COUNT[1:9]; open(70);find(100,[DG010002]); find(101,[DG010003]);
write text(0,[[cc]PART **IV[cc]]);
read binary(101,COUNT,[COUNT]);
nsor:=COUNT[1]; acont:=COUNT[2]; sgn:=COUNT[3]; pt:=COUNT[4]; cols:=COUNT[5]; zero:=COUNT[7];
if COUNT[6]=1 then f:=format([s-ndddddd.ddddd]);
if COUNT[6]=0 then f:=format([s-d.dddddd,nd]);
ax:=COUNT[8]; ay:=COUNT[9];

n:=0; rewind(101);

begin array ZEROS[1:zero],RESULT[1:pt],NODE,TYPE[1:nsor],T[1:acont],Q[1:acont];
for i:=1 step 1 until acont do
begin array t[1:1]; read binary(101,t,[t]);
T[i]:=t[1];
begin array PAIR[1:2];
for j:=1 step 1 until t[1] do
begin n:=n+1;read binary(101,PAIR,[PAIR]);
NODE[n]:=PAIR[1]; TYPE[n]:=PAIR[2];
end;
end;

end;

end;

```

```

rewind(101); read binary(101,ZEROS,[ZEROS]);
read binary(100,RESULT,[RESULT]);
if (sgn=1 and ax=1) or (sgn=-1 and ax=0) then
  begin for i:=1 step 1 until (acont+2) do
    begin k:=T[i]; T[i]:=T[acont+1-i]; T[acont+1-i]:=k;end;

    for i:=1 step 1 until (nsor+2) do
      begin k:=NODE[i]; NODE[i]:=NODE[nsor+1-i]; NODE[nsor+1-i]:=k;
        k:=TYPE[i]; TYPE[i]:=TYPE[nsor+1-i]; TYPE[nsor+1-i]:=k;
      end;
    n:=1;
    for i:=1 step 1 until acont do
      begin for j:=1 step 1 until (T[i]+2) do
        begin k:=NODE[n+j-1]; NODE[n+j-1]:=NODE[T[i]+n-j]; NODE[T[i]+n-j]:=k;
          k:=TYPE[n+j-1]; TYPE[n+j-1]:=TYPE[T[i]+n-j]; TYPE[T[i]+n-j]:=k;
        end;
        n:=n+T[i];
      end;
    n:=0;
    for i:=1 step 1 until acont do
      begin Q[i]:=0;
        for j:=1 step 1 until T[i] do
          if TYPE[j+n]≠0 then Q[i]:=Q[i]+1;
          n:=n+T[i];
        end;
        for i:=1 step 1 until (pt+2) do
          begin r:=RESULT[i]; RESULT[i]:=RESULT[pt+1-i]; RESULT[pt+1-i]:=r; end;
          n:=1;
          for i:=1 step 1 until acont do
            begin for j:=1 step 1 until (Q[i]+2) do
              begin r:=RESULT[n+j-1]; RESULT[n+j-1]:=RESULT[Q[i]+n-j]; RESULT[Q[i]+n-j]:=r; end;
              n:=n+Q[i];
            end;
            n:=0;
            for i:=1 step 1 until acont do
              begin Q[i]:=0;
                for j:=1 step 1 until T[i] do
                  if TYPE[j+n]=0 then Q[i]:=Q[i]+1;
                  n:=n+T[i];
                end;
                for i:=1 step 1 until (zero+2) do
                  begin r:=ZEROS[i]; ZEROS[i]:=ZEROS[zero+1-i]; ZEROS[zero+1-i]:=r; end;
                  n:=1;
                  for i:=1 step 1 until acont do
                    begin for j:=1 step 1 until (Q[i]+2) do
                      begin r:=ZEROS[n+j-1]; ZEROS[n+j-1]:=ZEROS[Q[i]+n-j]; ZEROS[Q[i]+n-j]:=r; end;
                      n:=n+Q[i];
                    end;
                    end;
                    end;
                    sgn:=0;

```

AGAIN:

```

if ax=1 and ay=0 then begin
oldnode:=newnode:=oldrslt:=newrslt:=oldzero:=newzero:=0;
min:=NODE[1];
for j:=2 step 1 until nsor -sgn do
if NODE[1]>NODE[j] and min>NODE[j] then min:=NODE[j];
for i:=1 step 1 until acont do
begin oldnode:=oldnode+1; n:=0;
for j:=1 step 1 until cols do
begin if T[i]≠0 then
begin if min-1+j=NODE[oldnode] then
begin if TYPE[oldnode]≠0 then begin oldrslt:=oldrslt+1;
write(70,f,RESULT[oldrslt]);
end;
if TYPE[oldnode]=0 then begin oldzero:=oldzero+1;
write(70,f,ZEROS[oldzero]);
end;
sgn:=sgn+1; oldnode:=oldnode+1; goto L1;
end;
end;
n:=n+1; space(70,14);
if j=cols then oldnode:=oldnode-1;
end;
if T[i]≠0 then begin T[i]:=T[i]+n-cols; if T[i]<0 then T[i]:=0; end;
for k:=1 step 1 until T[i] do
begin oldnode:=oldnode+1; newnode:=newnode+1;
NODE[newnode]:=NODE[oldnode]; TYPE[newnode]:=TYPE[oldnode];
if TYPE[oldnode]≠0 then begin oldrslt:=oldrslt+1; newrslt:=newrslt+1;
RESULT[newrslt]:=RESULT[oldrslt];
end;
if TYPE[oldnode]=0 then begin oldzero:=oldzero+1; newzero:=newzero+1;
ZEROS[newzero]:=ZEROS[oldzero];
end;
end;
newline(70,1);
end;
for i:=1 step 1 until acont do
if T[i]≠0 then begin newline(70,5);goto AGAIN; end;
end;
sgn:=0;
if ax=0 and ay=1 then
begin n:=1;
for i:=1 step 1 until acont do
begin for j:=1 step 1 until (T[i]÷2) do
begin k:=NODE[n+j-1]; NODE[n+j-1]:=NODE[T[i]+n-j]; NODE[T[i]+n-j]:=k;
k:=TYPE[n+j-1]; TYPE[n+j-1]:=TYPE[T[i]+n-j]; TYPE[T[i]+n-j]:=k;
end;
n:=n+T[i];
end;
end;

```

L1:

```

n:=0;
for i:=1 step 1 until acont do
begin Q[i]:=0;
  for j:=1 step 1 until T[i] do
    if TYPE[j+n]≠0 then Q[i]:=Q[i]+1;
  n:=n+T[i];
end;
n:=1;
for i:=1 step 1 until acont do
begin for j:=1 step 1 until (Q[i]÷2) do
  begin r:=RESULT[n+j-1]; RESULT[n+j-1]:=RESULT[Q[i]+n-j]; RESULT[Q[i]+n-j]:=r; end;
  n:=n+Q[i];
end;
n:=0;
for i:=1 step 1 until acont do
begin Q[i]:=0;
  for j:=1 step 1 until T[i] do
    if TYPE[j+n]=0 then Q[i]:=Q[i]+1;
  n:=n+T[i];
end;
n:=1;
for i:=1 step 1 until acont do
begin for j:=1 step 1 until (Q[i]÷2) do
  begin r:=ZEROS[n+j-1]; ZEROS[n+j-1]:=ZEROS[Q[i]+n-j]; ZEROS[Q[i]+n-j]:=r; end;
  n:=n+Q[i];
end;
for i:=1 step 1 until (acont-sgn) do
begin sum:=1; v:=sgn;
  n:=0;
  for k:=1 step 1 until (if acont-sgn<cols then acont-sgn else cols) do
    n:=n+T[k+sgn];
  min:=NODE[1];
  for j:=2 step 1 until n do
    if NODE[1]<NODE[j] and min<NODE[j] then min:=NODE[j];
  for j:=1 step 1 until (if acont-sgn<cols then acont-sgn else cols) do
  begin v:=v+1; if T[v]=0 then goto LY1;
    if min>NODE[sum] then goto LY1;
    if min=NODE[sum] then
      begin n:=m:=0; for k:=1 step 1 until v-1 do n:=n+T[k];
        if TYPE[sum]≠0 then
          begin for k:=1 step 1 until n do
            if TYPE[k]≠0 then m:=m+1;
            write(70,f,RESULT[m+1]);
            for k:=1 step 1 until (pt-m-1) do
              RESULT[m+k]:=RESULT[m+k+1]; pt:=pt-1;
          end;
          if TYPE[sum]=0 then
            begin for k:=1 step 1 until n do
              if TYPE[k]=0 then m:=m+1; write(70,f,ZEROS[m+1]);
              for k:=1 step 1 until (zero-m-1) do
                ZEROS[m+k]:=ZEROS[m+k+1];
            end;
        end;
      end;
    end;
  end;
  LY1:
end;

```

JJ:


```

                                zero:=zero-1;
                                end;
                                T[v]:=T[v]-1; if T[v]<0 then T[v]:=0;
                                newrslt:=newzero:=0;
                                for k:=v step 1 until acont do newrslt:=newrslt+T[k];
                                for k:=1 step 1 until (v-1) do newzero:=newzero+T[k];
                                for k:=1 step 1 until newrslt do
                                begin NODE[newzero+k]:=NODE[newzero+k+1];
                                    TYPE[newzero+k]:=TYPE[newzero+k+1]; end;
                                sum:=sum+T[v]; goto LY2;
                                end;
LY1:                                sum:=sum+T[v]; space(70,14);
LY2:                                end;
                                newline(70,1); v:=sgn;
                                for k:=1 step 1 until (if acont-sgn<cols then acont-sgn else cols) do
                                if T[v+k]≠0 then begin sum:=1; goto JJ; end;
                                sgn:=sgn+cols; newline(70,5);
                                end;
                                end;
                                end;
                                close(70);close(100);close(101);
end→

```

INDEX OF REFERENCES

1. Allen , D.N.de G(1954) Relaxation methods ,McGraw hill, New York , London.
2. Allen Fox, southwell (1946) Relaxation method applied to engineering problems. Phil.Trans. Roy. Soc. A,239, 501-537.
3. Appl. Math. Ser. U. S. Bur. Stand. 37(1954) Tables of functions and of zeros of functions, Government printing press, Washington.
4. Bickley, .G.(1948) Finite difference formulae for the square lattice, Quart . J. Mech. Appl. Maths. Vol. 1,35-42.
5. Collatz, L(1959) Numerical treatment of differential equations (Translated by P.G. Williams) Springer-Verlag, Berlin.
6. Forsythe and Wasow (1960) Finite difference methods for partial differential equations, John Wiley, New York.
7. Fox , L (1950) Numerical solution of elliptic differential equations when the boundary conditions involve a derivative , Phil. Trans. Roy. Soc. A , 242,345-378.
8. Fox, L(1962), [editor], Numerical solution of ordinary and partial differential equations-Oxford Summer School, Pergamon Press, London.
9. Greenspan, D(1965) Introductory numerical analysis of elliptic boundary value problems, Harper and Row , London.
10. Jeffreys, H and Jeffreys, B.S (1950) methods of mathematical physics, Cambridge University Press.
11. KDF 9 ALGOL manual, publication 1002, English Electric Leo computers Limited, Kidsgrove, Staffs.
12. Milne, W.E(1957) Numerical solution of differential equations, John Wiley , New York, London.
13. Motz, H (1946) the treatment of singularities of partial differential equations by relaxation methods; Quart. Appl. Math.4;372-377.

14. Scarborough, J.B.(1958) Numerical mathematical analysis; Baltimore : The Johns Hopkins Press; Oxford University Press.
15. Shaw, F.S(1950) An introduction to relaxation methods; Dover publications, New York.
16. Sneddon , I.N(1962a) Note on an electrified circular disk situated inside a coaxial infinite hollow cylinder; Proc. Camb. Phil. Soc.; 58;621-624.
17. Southwell, R.V.(1942) Proc. Roy. Soc. A, 180;367-396.
18. Srivastav, R.P(1963) Certain dual series relations and their applications; Ph.D. thesis, Glasgow University.
19. Varga, R.S(1957) Numerical solution of the two group diffusion equation in x-y geometry ; Inst. Radio Engrs.; Trans. Prof. Group Nuclear Science, Vol. NS-4,52-62.
20. Vishwanathan , R.V(1957) solution of Poisson's equation by relaxation methods-normal gradient prescribed on curved boundaries ; Math. Tables Aids Compt. ,Vol.11. 67-78.
21. Wilson, S.B.L(1963) A difference method for the numerical solution of boundary -value problems; Ph.D. thesis, Glasgow University.
22. Woods, L.C(1953) The relaxation treatment of singular points in Poisson's equation; Quart. J. Mech. 6.163-185.

